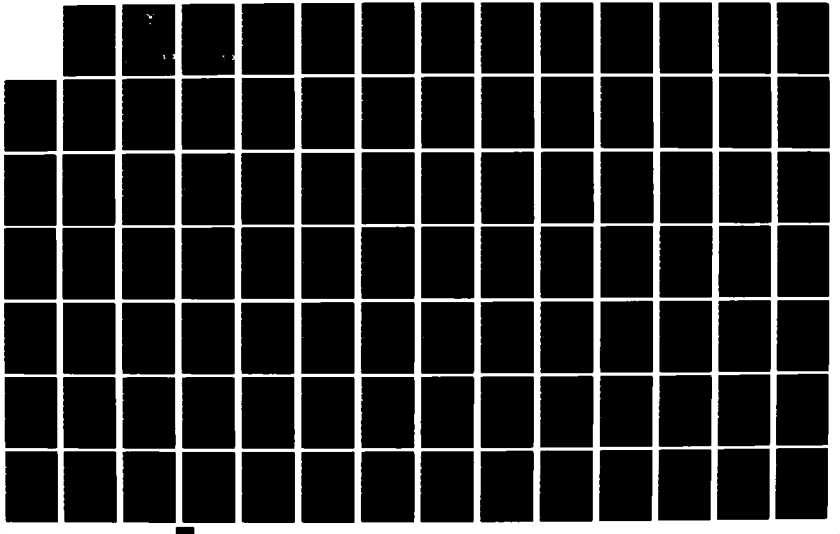
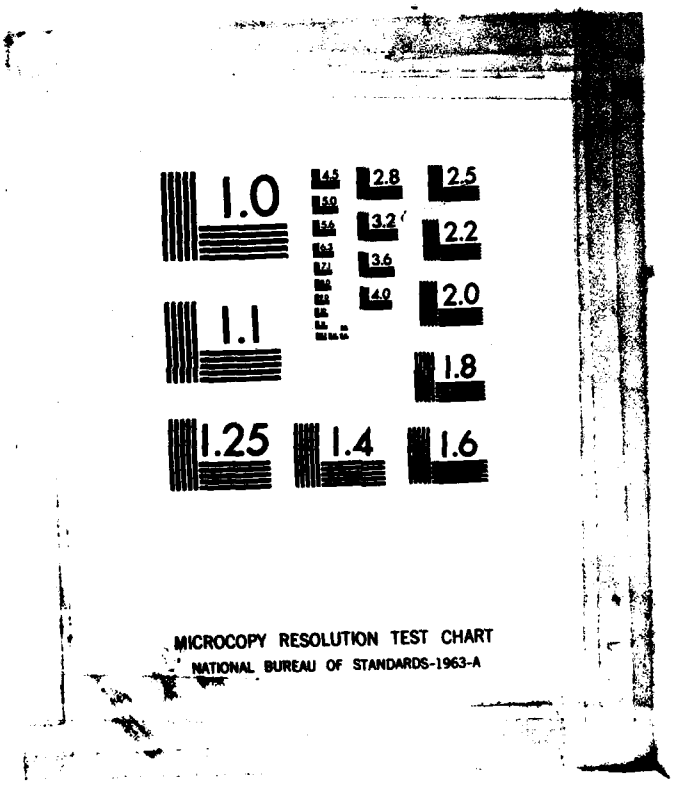
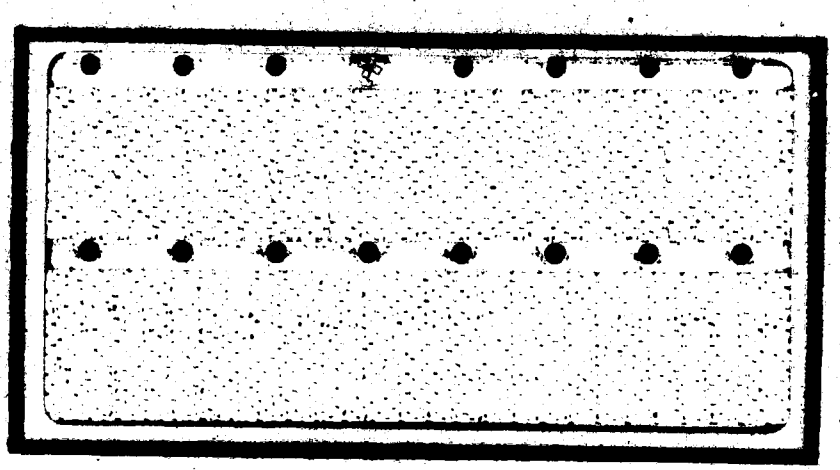
AD-A124 834 DESIGN OF A PROTOTYPE UNIVERSAL NETWORK INTERFACE DEVICE USING INTEL 8086.. (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. D E PALMER 1/3

UNCLASSIFIED DEC 82 AFIT/GE/EE/82D-52 F/G 9/2 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/EE/82D-52

DESIGN OF A PROTOTYPE UNIVERSAL NETWORK INTERFACE DEVICE USING INTEL 8086 AND 8089 16-BIT MICROPROCESSORS

THESIS

AFIT/GE/EE/82D-52 Donald E. Palmer
1Lt USAF

Approved for public release; distribution unlimited.

AFIT/GE/EE/82D-52

DESIGN OF A PROTOTYPE UNIVERSAL NETWORK INTERFACE DEVICE USING INTEL 8086 AND 8089 16-BIT MICROPROCESSORS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Donald E. Palmer
1Lt, USAF
Graduate Electrical Engineering
December 1982

Preface

This research effort describes the prototype development of an improved Universal Network Interface Device (UNID II). The UNID II's architecture was based upon a preliminary design project at the Air Force Institute of Technology. The UNID II is comprised of two modules; a local module and a network module. The operations of both modules are controlled with 16-bit 8086 microprocessors. The network utilizes an additional 8089 Input/Output Processor for controlling I/O operations. This report documents the detailed design, construction, and testing of the UNID II's network module. Tests performed on the local module and the shared memory are also documented.

I would like to thank my thesis advisor, Dr. Gary Lamont, for his assistance and encouragement throughout the course of this investigation. I would also like to thank my readers, Major W.D. Seward and Major C.W. Lillie, for their valuable comments and aid during this project. The excellent technical support by the laboratory technicans was greatly appreciated. I wish to thank Mr. Orville Wright and Mr. Dan Zambon for their assistance. Finally, I wish to express my deepest appreciation to my wife, Phyllis, for her encouragement, assistance, and understanding during my entire graduate program.

Contents

List of Figures

List of Tables

Abstract

This research describes the development of a Universal Network Interface Device (UNID II) which is intended to function as a network node in a computer communications network. The UNID II is a 16-bit, 8086 microprocessor based version of the present 8-bit Z80A UNID being developed at the Air Force Institute of Technology (AFIT). The UNID II's architecture was based on a conceptual block diagram design presented in a previous AFIT thesis. It is comprised of two modules: a local module, which interfaces the UNID II to a host computer and peripheral devices; and a network module, which interfaces the UNID II to a computer communications network. In this report the detailed design, construction, and testing of the network module is documented. The network module was designed using the Intel 8086 microprocessor family of components, including an 8089 Input/Output processor. An Intel SBC 86/12A single board computer was used as the local module and its testing is also documented. The tests were conducted with the aid of an Intel ICE-86A/88A In-Circuit Emulator. The tests conducted, verified the proper operation of the network module's bus interface circuitry, control circuitry, and DMA transfer capabilities. The shared memory, which is used for intercommunication between the two modules, was also successfully tested. The UNID II was not tested in a computer communications network environment.

I Introduction

Local networks (often called local area networks (LANs) or local computer networks (LCNs), are generally referred to as data communication networks which interconnect computers and terminals over a limited geographical area (Ref 31:18). In this thesis, the development of a LCN interface device is presented. This Universal Network Interface Device (UNID II) design is based on the 16-bit architecture of the Intel 8086 microprocessor. The Intel 8086 family of processors was selected for implementation of the UNID II because of its bus support circuitry which eases the development of multiprocessor systems (Ref 14:21). Also, of the three 16-bit microprocessors (Intel 8086, Motorola 68000, and Zilog Z8000), the 8086 has the most compact instruction format for expressing the different addressing modes. the most extensive set of byte-data arithmetic features, and it is the only processor of the three which allows byte data to lie on odd addressed locations (Ref 13:157). These three features are desirable for the UNID implementation since it functions more as a string data manipulator than as a number cruncher.

Historical Perspective

Local computer networks have evolved from the large, long-haul, distributed processing networks developed in the 1960's and early 1970's such as Arpanet, Tymnet, GE Information Services, and others (Ref 38). The recent interest

in LCNs has been spa ed by the rapid advancements made in semiconductor technology. Increasing the density of semiconductor components on integrated circuits has enabled electronic computer components to be made physically smaller, to operate faster, and to cost less. Some microprocessors presently being manufactured have surpassed the computing speed and memory addressing capabilities of the large, room-sized computers of two decades ago, and at a fraction of the cost (Ref 3:98). The microprocessor improvements in conjunction with advancements in peripheral components, such as intelligent terminals and secondary storage disk drives, have enabled the desk-sized computer to become a reality. As a computer's components become cheaper and their performance factors increase, it is becoming more economical for computing power to be dispersed among many of the smaller desk-sized computer systems (Ref 3:96). The LCN will function as a communication interface to interconnect these computers, terminals, and possibly mainframe type computers to form a network for expanded resourse sharing.

Background

The first consideration given to the development of an LCN for the Air Force Institute of Technology's (AFIT's) Digital Engineering Laboratory (DEL) was presented in a 1978 thesis (Ref 36). In this thesis, the preliminary requirements for the network (termed DELNET) were defined. Also, during that same year, work was begun on a Universal Network Interface Device (UNID) which would permit the DELNET and

various host computers and peripherals to be interconnected (Ref 39). This work was based on the need for such a device as proposed in a technical report authored by the 1842 Electronic Engineering Group (EEG) of the Air Force Communication Services (Ref 1). In this technical report, the 1842 EEG recommended that local networks connected in a multi-ring configuration be used to upgrade base operations and telecommunication capabilities at Air Force bases. The multi-ring network they presented, required five different types of interface devices for interconnecting the different network rings and for connecting the users to the network. In the 1978 AFIT thesis, the functional requirements necessary for one UNID to perform the tasks of the five types of network interface devices were defined and a prototype device was partially designed.

In 1979, the hardware development of a prototype UNID was initiated (Ref 5). This development was based on the UNID design formulated in the previous thesis effort. The hardware of this prototype consisted of a local input/output (I/O) card, network I/O card, local and network Z80 processor cards, shared memory card, and a dual processor card which arbitrates the shared memory between the two processors and controls the memory refresh circuitry. The software developed during this thesis was limited to simple routines for testing the UNID.

In 1980, the hardware design and the testing of the prototype UNID was completed (Ref 2). The operating system's

software for the local and network modules was also developed. The UNID monitor was enhanced to provide UNID software debugging support and to facilitate the downloading of programs from a MCZ1/25 development system (Ref 46) to the UNID. Also, the UNID's memory arbitration circuitry was redesigned and the dual processor card was eliminated. The UNID now consisted of the following circuit boards:

- Local Input/Output Board (4 serial ports)
- Network Input/Output Board (2 serial ports)
- Local Processor Board
- Network Processor Board
- Shared Memory Board
- System Memory Board

In March 1981, the system requirements and design of the DELNET were redefined to include the requirements of the DELNET users, as well (Ref 16). The initial DELNET topology specified during this project is shown in Figure 1. The network nodes (UNIDs) are connected in a basic loop configuration with a star type topology being used for connecting the hosts to the network nodes.

The loop-type network was selected for the initial DELNET design because its relatively simple routing procedures would help to reduce network development time and the network could easily be expanded as new nodes were constructed. Two disadvantages to this type of topology are that if one node of the network fails, then the entire network is rendered inoperative, and as the number of nodes in the network increases, the network response time also increases due to an increase in the transient delay time for

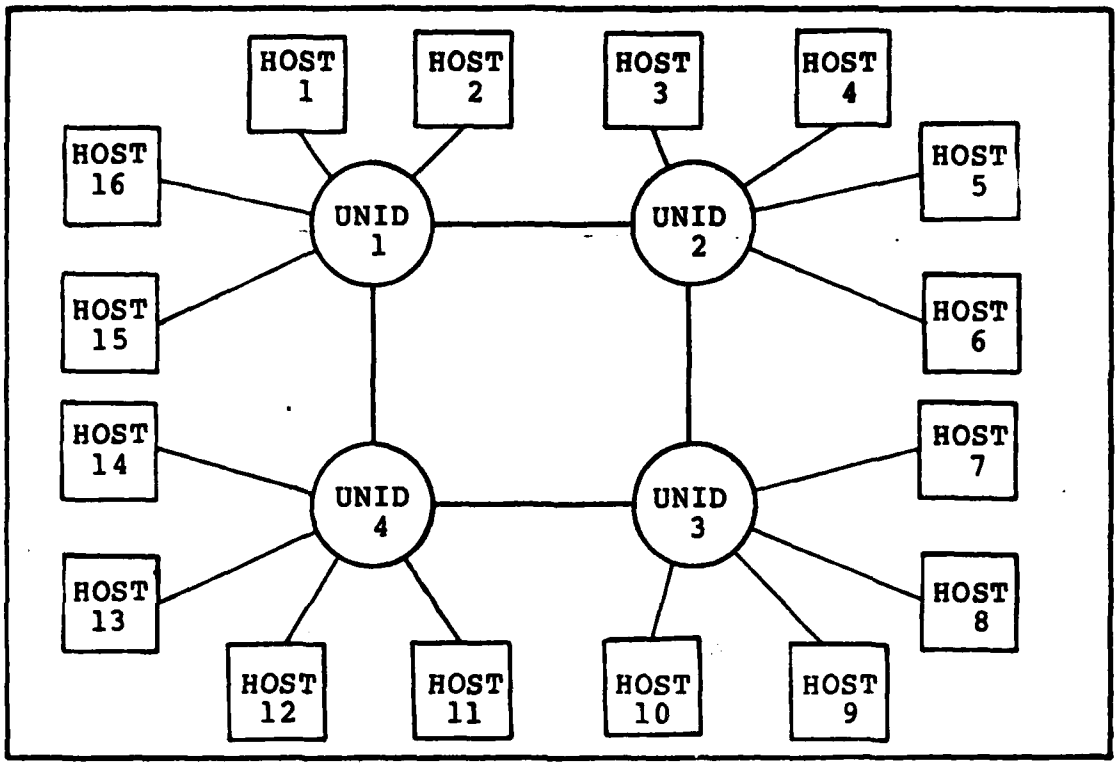


Figure 1. DELNET Topology (Ref 16)

each message. These disadvantages, however, did not appear to be a problem. The star topology connecting the hosts to the nodes would help to minimize the number of nodes required in the network. Also, complete availability of the network was not considered to be crucial (Ref 16:83-85).

Two thesis projects were undertaken during the latter part of 1981. In one, the development of the software for the DELNET was continued (Ref 12) and in the other, the construction of two UNID prototypes was completed (Ref 35). A demonstrative testing of the UNID in a partial LCN configuration (two nodes) was also performed.

Also in 1981, work began on the designing of an improved UNID model (UNID II) (Ref 14). The UNID II was

designed to upgrade the UNID from its 8-bit configuration to a faster more powerful 16-bit architecture using Intel 8086 microprocessors and an 8089 Input/Output processor. The above work is the foundation upon which this project is based.

Problem and Scope

As mentioned previously, several thesis projects have been performed over the past few years which were concerned with the development of a local computer network for the AFIT digital engineering laboratory. A major problem that is confronted when developing any LCN involves the interfacing of many different types of equipments to the network. The network interface device should be designed for flexibility in order to perform the tasks required to function as a network node, and for interfacing to the different types of equipments. Also, the operation of the network interface device should be transparent to the user.

This study focused on the construction and implementation of a Universal Network Interface Device using 8086 family components. These components have been designed to operate together to facilitate the development of microcomputing systems which can be specifically tailored to meet the needs of a particular application without requiring excessive and unnecessary capabilities also being included (Ref 20:1-1). This UNID is referred to as a UNID II since it is an upgraded model from the UNID's previous 8-bit, Z80A based design.

A block diagram of the UNID II is shown in Figure 2, it consists of two modules, a network module for interfacing the UNID II to the computer network, and a local module for interfacing the UNID II to the user's equipments. An off-the-shelf Intel SBC 86/12A single-board computer (Ref 19) was used for the local module. Since no off-the shelf boards containing an 8089 IOP were available, it was necessary that the network module be designed and constructed. The two modules communicate with each other through a block of shared memory located on the local module. All communication between the two modules takes place over an Intel Multibus (Ref 20:A-175) which functions as the system bus.

Software test algorithms were developed during this project to aid in the testing and validation of the UNID II. The algorithms which will enable the UNID II to function as a network node for the DELNET are being developed in a concurrent thesis project (Ref 15). These algorithms are being developed using PL/Z high-order-language (Ref 47) and they will need to be converted to PL/M 86 (Ref 28) for implementation in the UNID II. This algorithm conversion is not within the scope of this effort.

Approach

The first task in developing the UNID II involved conducting a literature search to find current information concerning local networks and their device interface configurations. Familiarity with the operation of the 8086 and 8089 microprocessors was gained by studying the manufac-

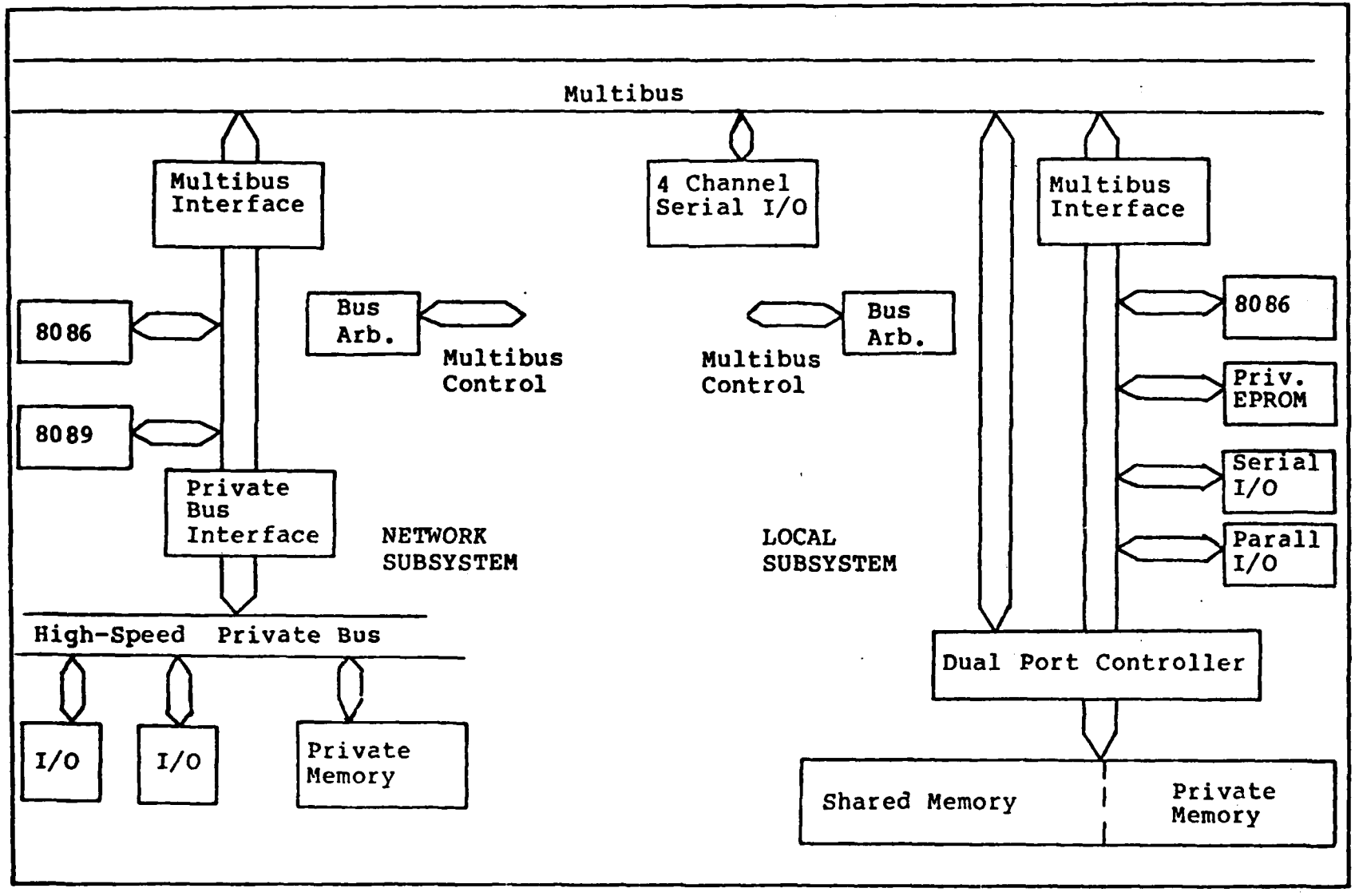


Figure 2. UNID II Block Diagram

turer's literature and by experimenting with the 8086 computer systems avaiable in the lab.

Before constructing the UNID II as configured in the block diagram (Figure 2), it was decided that a demonstration prototype of the network module be designed and constructed. This decision was based upon two primary factors: First, the Signetics 2652 Multi-protocol Communication Controller (MPCC) integrated circuits (ICs) (Ref 18) being used for the Network I/O ports were on order and they were not expected to be received for three or four months. Second, the demonstration network module was to be identical to the network module, except that an 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) was to be used as a network I/O port instead of the MPCCs. By using a familiar device, such as the USART for an I/O port, the testing and validation of the bus interface circuitry and other network module components could be completed before having to be concerned with the initilization and programming of the complex MPCC devices.

The actual design of the UNID II was performed in stages. First, the network module containing an 8086 microprocessor and the 8089 I/O processor was designed and its bus interface circuitry was breadboarded. (The details concerning the design, construction, and testing of this board are covered in later chapters of this report). After the bus interface circuitry was tested and its proper operation verified, a wire-wrapped prototype of the

demonstration module was constructed. (A wire routing computer program was utilized to obtain a wiring list which aided in the construction of this board. A description of the wire routing program and the wiring list for the demonstration module are located in Appendix A). The network board was then checked for wiring errors and applied power tests were performed.

Next, the local module was tested. Since the Intel SBC 86/12A was being used as the local module, no major testing problems were anticipated or encountered. An In-Circuit Emulator for an 8086 (ICE-86A)(Ref 24) was used to exercise the SBC 86/12A to verify its operation. The ICE-86A was then used to test the operation of the network module's resident bus interface circuitry. Both modules were then interfaced together over the Multibus and the ICE-86A was utilized to test the system bus interface and arbitration circuitry. The proper operation of shared memory and the ability of the 8089 to perform DMA (direct memory access) transfers were also verified.

After the demonstration network module was successfully tested, the interface circuitry for controlling the two 2652 MPCC ICs was designed. These two chips serve as the serial I/O ports which are to interface the network module to the DELNET.

The software for testing the UNID II was developed on an Intel Intellec Series II Microcomputer Development System (MDS) (Ref 26) using Intel's PL/M 86 general purpose, high-

order language (Ref 28).

Overview of Thesis

The structure of this report conforms to the procedures specified in the approach. Chapter I covers background information previous to this development and the approach undertaken in designing and constructing the UNID II prototype. Chapter II contains a summary of the UNID and DELNET requirements analysis, which serve as a basis for the UNID II design. The remaining chapters follow the basic breakdown of the design steps as listed in the Approach. Chapter III describes the development of network interface module. Chapter IV presents the procedures of testing the UNID II. Chapter V contains an overall thesis summary with recommendations for further study and development.

II. UNID II Requirements

This chapter summarizes the requirements established in previous thesis projects. First, a summary of the UNID and UNID II requirements are presented. Then the DELNET requirements are summarized. Next, the UNID/DELNET protocol requirements are discussed and an overview of the standards to be implemented by UNID II are presented.

UNID Requirements Summary

The design of the original UNID was based on the following general concepts (Ref 39:13):

- The UNID should function as a store-and-forward concentrator and have message routing capabilities.
- The UNID might require specalized I/O ports for unique communication requirements.
- The UNID should be capable of interfacing to various network operating systems and protocols.

These concepts are still primary design factors for the UNID II. As mentioned previously (Chapter 1), the proposed DELNET configuration is a combination ring and star type topology. However, the UNID should be designed so that it can easily be connected into other types of network configurations. It is also important for the UNID to be hardware independent of the network protocols and routing procedures used. This would ensure that changes in the network-to-UNID environment could easily be acheived through modifications of the UNID's software, not its hardware.

At the lowest levels of protocol, the UNID-to-computer, UNID-to-peripheral, and UNID-to-network interfaces have been defined to conform to the EIA RS-449 (Ref 8) and RS-232C (Ref 7) communication standards (as explained later in this chapter). However, at the higher levels of protocol, which perform the message routing, message processing, flow control, and other functions, the UNID should be transparent. That is, the UNID should not impair the functioning of these higer levels, and any change at the higher protocol levels should be implementable by modifications of software only.

The three basic concepts and structured analysis techniques were used to define the original UNID functional requirements (Ref 39). The final UNID design was developed using a modular design approach which identified three seperate modules: (1) a local Input/Output (I/O) module for interfacing the UNID to the user's peripherals or modems; (2) a network I/O module for interfacing the UNID to the network; and (3) a dual processor module for matching the UNID's throughput to the network environment (Ref 39:154-155). These three types of modules were selected after defining the UNID functional requirements with the aid of the Structured Analysis Design Technique (SADT)(Ref 39:11-31).

In 1981 a thesis project was initiated to design an improved UNID (UNID II), and the original UNID functional requirements were reevaluated (Ref 14:17-35). Using Data

Flow Diagrams (Ref 43), a functional requirements model for UNID II was generated. This functional model categorized the UNID II functional requirements into two main groups; one for handling local messages and one for handling network messages. The data flow diagrams of the model indicated that the operations of the two functional groups were very similar, but both were necessary to process both network and local messages. The input requirements for UNID II which were specified by the model are listed in Table I. The functional requirements model serves as a basis for the UNID II design and these diagrams are of sufficent detail to provide guidelines for the implementation of the UNID II. The data flow diagrams of the model are presented in Appendix A.

DELNET Functional Requirements

An evaluation of the functional requirements of the DELNET was performed in a 1981 thesis project (Ref 15:19-23). Responses from a three part user survey were used to formulate these functional requirements. A summary of the most important requirements which were defined are listed below:

- Ability to transfer files accross the network.
- Ability to share peripherals attached to the hosts on the DELNET.
- Flexibility with respect to the network topology protocols, and transmission medium used.
- Performance monitoring capability.
- High percentage of availability.

Table I. UNID II Input Requirements

UNID II Input Requirements:

I. Interface a wide variety of network components and handle various topologies.

- A. Accommodate dissimilar computing equipment
  - 1) Accomplish code conversion
  - 2) Perform data-rate speed conversion
- B. Interface peripherals and user terminals to network
- C. Interface host computers to network
- D. Provide a network-to-network interface (gateway)

II. Perform independently of network components

- A. Handle network data transmission and reception
  - 1) Accommodate network throughput requirements
    - a) Provide flow control
  - 2) Adaptable to different protocols
    - a) Handle both synchronous and asynchronous communication
    - b) Edit and pack characters into formatted message
    - c) Unpack a message
    - d) Perform Serial to parallel data conversion
    - e) Handle error control functions such as Message Acknowledge, No Acknowledge, Repeat, and Timeout
  - 3) Have error checking and recovery capability
- B. Relieve host computers from network specific functions
  - 1) Provide a buffer to smooth message traffic
  - 2) Poll communication lines if they are multidropped
  - 3) Handle Interrupts
  - 4) Route messages to desired destination
  - 5) Collect performance, traffic, and error statistics

- User transparency to network configuration and specific operating systems of hosts.

Other requirements were defined from the user survey, but they were not considered to be of immediate concern. These requirements included the following: ability to perform distributed processing, and work with distributed databases; permit software tool sharing; incorporate fault tolerance; provide gateways for connecting to the base CYBER 750 and other networks, such as ARPANET; and provide security for classified projects.

Based upon these user desired functional requirements, the DELNET hardware and software system requirements were defined. The DELNET topology selected was the ring-type discussed in the previous chapter (Figure 1). The use of this ring-type topology for the initial DELNET configuration will ease the development of routing algorithmns and allow for simple system expansion, since nc elaborate routing scheme is necessary and additional nodes or hosts can easily be connected into the network. The system requirements specified in this report were the building blocks used in the succeeding two thesis projects. In one of these projects, concentration was on the development of DELNET and UNID software procedures (Ref12). In the other, the continued development of two operational UNIDs was completed (Ref 35). Together they demonstrated the operation of the UNID in the partial DELNET terminal configuration shown in Figure 3, and the partial DELNET computer configuration

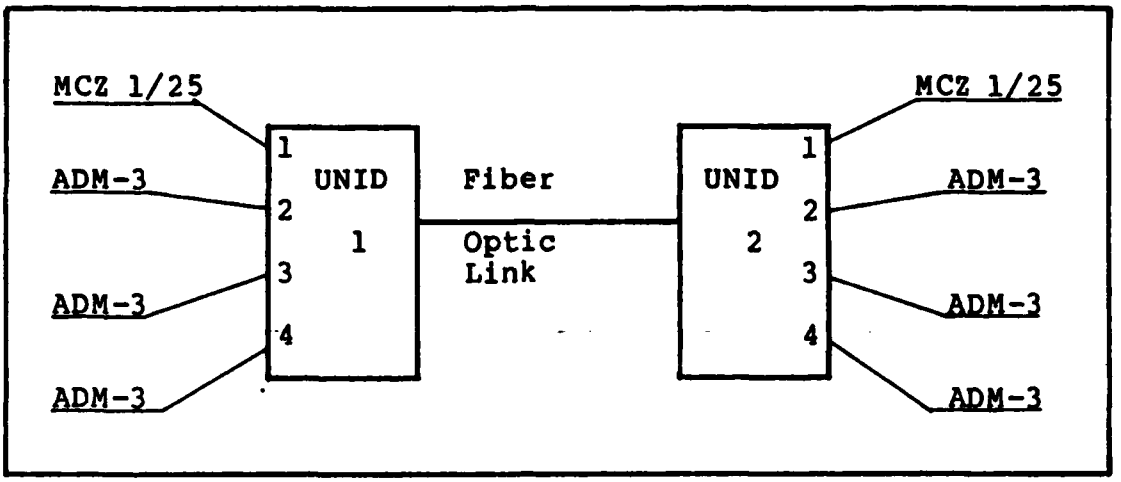


Figure 3. Prototype DELNET Terminal Configuration

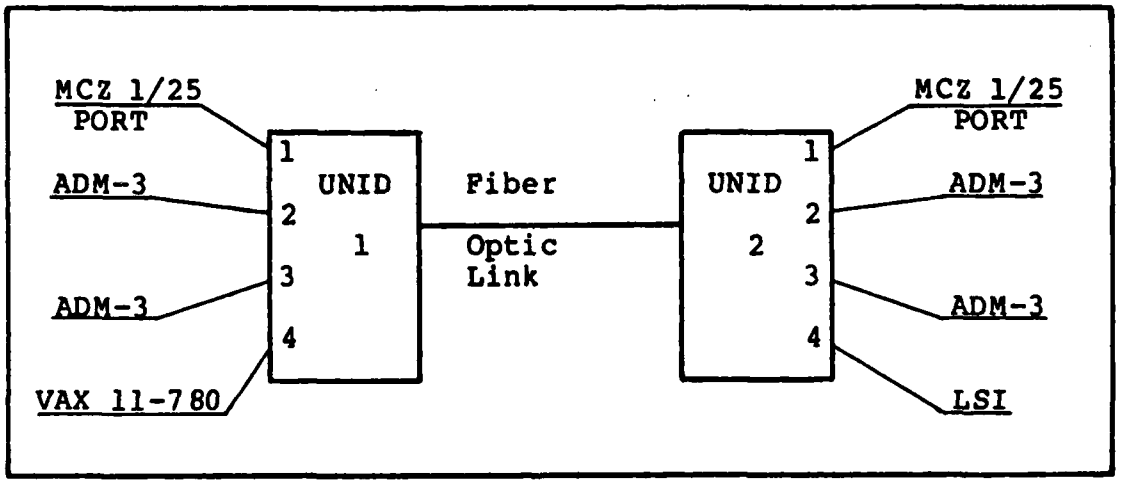


Figure 4. Prototype DELNET Computer Configuration

shown in Figure 4 (Ref 35:74).

Protocols

Protocols have been described by Weissberger as "...simply a set of rules that must be obeyed to ensure an orderly information exchange between two or more parties" (Ref 45:105). In specifying the "set of rules" requirements

for the DELNET, a packet-switching protocol was specified as being required and the X.25 protocol recommendation by the International Consultative Committee on Telephones and Telegraphs (CCITT) was selected (Ref 15:45).

Further evaluation of the protocol requirements resulted in the suggestion that the DELNET should use the Reference Model of Open Systems Interconnection (OSI) developed by the International Standards Organization (ISO) (Ref 12). The development of UNID and DELNET specific protocols is being undertaken in a concurrent thesis effort (Ref 15). However, since the UNID II is to be designed to function as a network node, it should support the lower three layers of the ISO model and a brief description of the ISO model and of the X.25 recommendation follows. Supportive information was obtained from several sources (Ref 4,6,10,33,41,48).

ISO Reference Model. The ISO Reference Model of Open System Interconnection is divided into the seven hierarchial layers shown in Figure 5. The highest three layers, the Application Layer, the Presentaion Layer, and the Session Layer are associated with the user and the host environment. The content of the Application layer is determined by the user and may include such functions as file transfers and execution of remote jobs. The functions of the Presentation layer and all of the lower layers are to provide support for the application layer (Ref 48:430). The Presentation layer performs all data format transformations and the services of

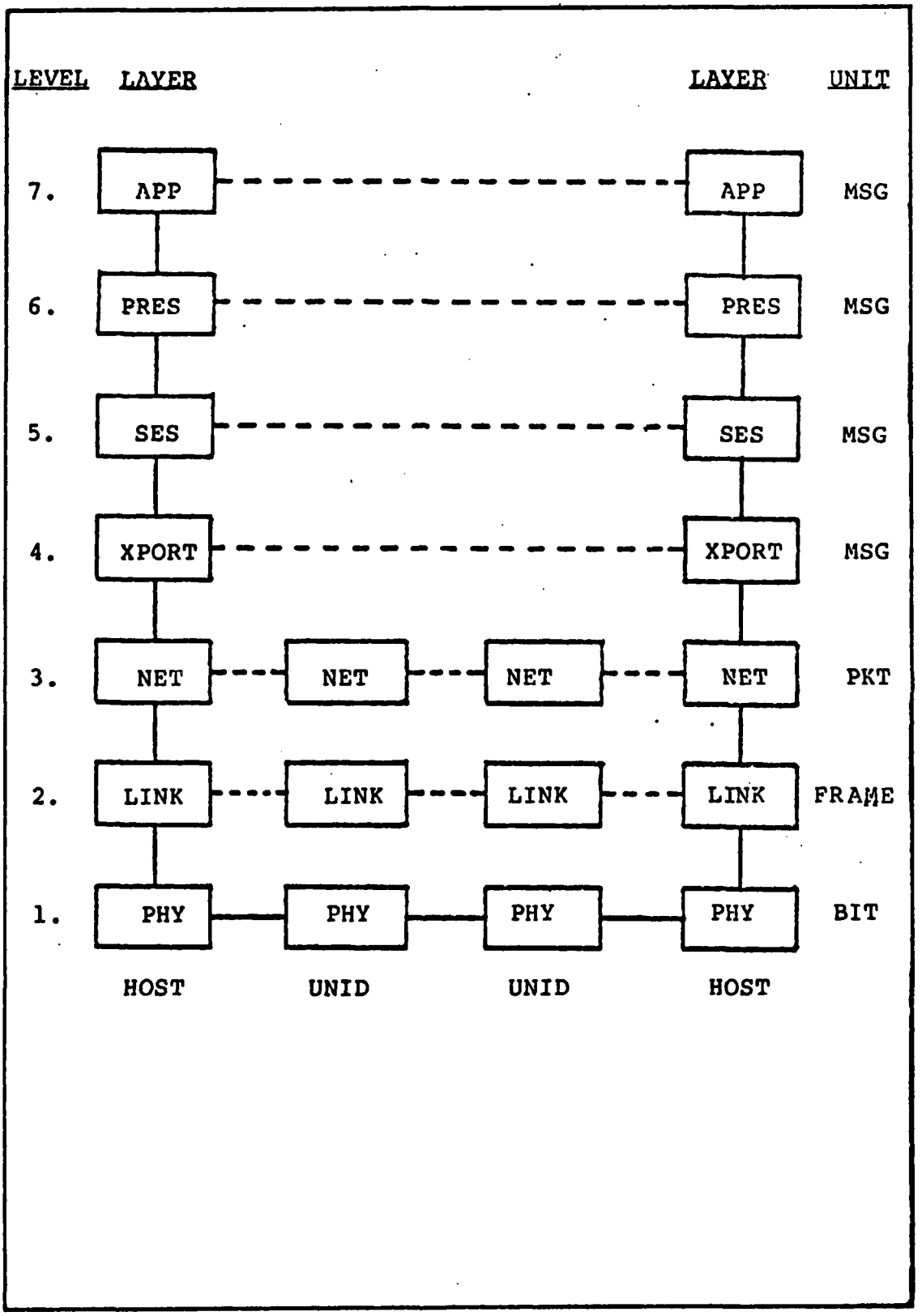


Figure 5. ISO Protocol Model with UNID (Ref 12:13)

the Session layer include addressing and connection management of the network.

The Transport layer services provide the host level data communcation facilities. It is sometimes referred to as the host-to-host layer. This layer must provide "reliable" and "efficent" transport control of messages between the host computers. Some examples of the services at this level include flow control, error recovery, and connection establishment/disestablishment.

The access protocol recommended for the lower three layers is the X.25 recommendation by the CCITT and the description of these three layers are included in the following discussion of the X.25 protocol.

X.25. The CCITT recommendation X.25 describes the interface and procedures for packet switched service. It is defined in three independent architectural levels which support the three lower layers of the ISO reference model (Ref 11). These three layers are presented in the following paragraphs.

Physical Layer. This layer is the lowest link in a network and it provides the physical, electrical, mechanical, functional, and procedural services to define the physical connection between data terminal equipments (DTE) and data circuit terminating equipments (DCE) (Ref 4). The physical standard referenced by the CCITT is the X.21 digital interface standard which is designed to interface a host computer to a network. In X.21, the host is specified

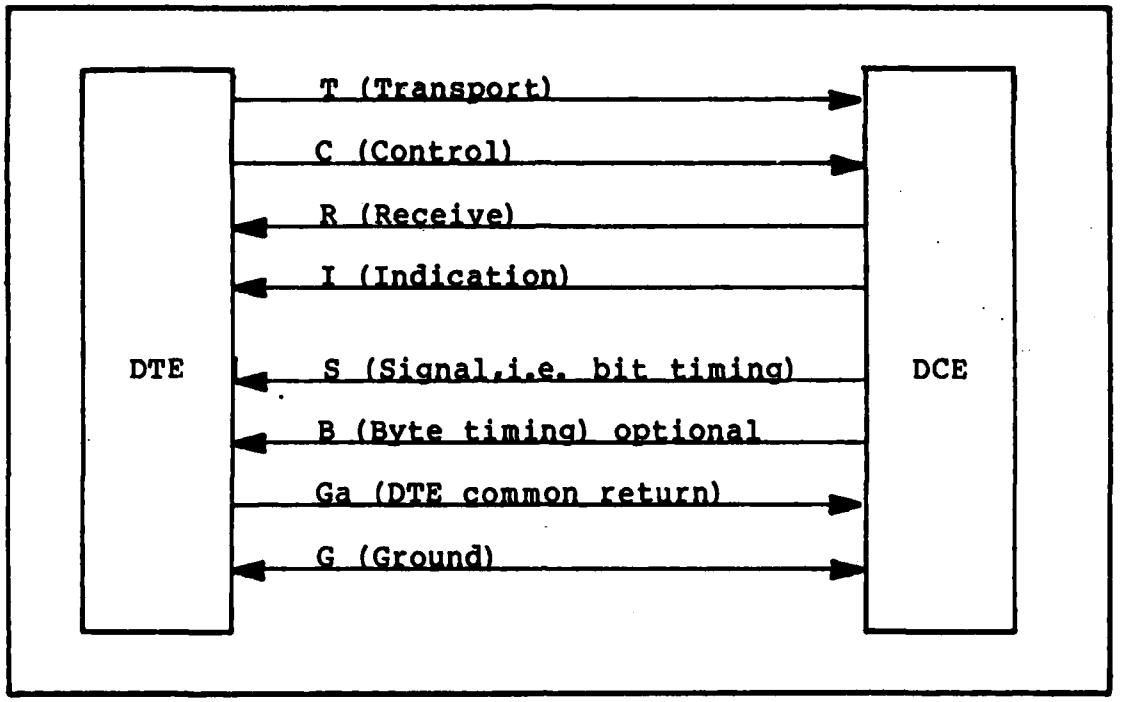


Figure 6. DTE/DCE Interface Connection in X.21 (Ref 41:109)

as the DTE and the network interface node is the DCE (Ref 40:461). The present physical layer standards, like RS-232C (Ref 7) and RS-449 (Ref 8), utilize analog signaling over the DTE/DCE interface.

Figure 6 shows the eight lines defined by X.21 for the connection of the DTE/DCE interface. The X.21 standard is concerned with the transmission of logical bits, so the "S" line provides the clock timing signal to define the bit boundaries and the "B" line (optional) allows for a timing pulse every eighth bit for byte alignment. The "T" and "R" lines are for the transmission of data and signaling information. The "C" and "I" lines are for control types of information. The X.21 connector has 15 pins but not all of

Flag	Address	Control	Infor-mation	FCS*	Flag
F 01111110	A 8-bits	C 8-bits	I N-bits	FCS 16-bits	F 01111110

* Note: FCS = frame checking sequence

Figure 7. X.25 Frame Structure (Ref 11:A8)

them are used.

Data Link Layer. Since the physical layer is only concerned with the transmission of bits over the DTE/DCE interface, it is the function of the Data Link Layer to create, recognize, and control the flow of the logical bits supplied to/from the physical layer. This layer combines the bits into logical units referred to as frames or packets. The bit-oriented link access control procedure specified in X.25 is the Link Access Procedure B (LAPB) which is equivalent to the ISO High Data Link Control (HDLC) standard (Ref 11:2). The frame format specified by this standard is shown in Figure 7.

Network Layer. This layer, referred to as the packet level by the CCITT, is concerned with the format and meaning of the data field contained within the frames (Ref 41:238). This layer's services include the routing and management of the data packets.

In the 1980 revision of X.25, a number of significant technical enhancements were made at this level. Two of the most important were; the addition of provisions for Datagram service, and the addition of a fast select facility to the

virtual call service of X.25 (Ref 11:2).

The Datagrams are self-contained packets which contain sufficent address information to be routed to their destinations and they may contain up to 128 bytes of user data. No set-up calls are required. The fast select facility provision allows a full 128 bytes of user data to be exchanged during the call set-up and clearing procedures for a virtual call (Ref 11:2). A more detailed description of these services can be found in the literature (Ref 41, 11).

Standards

All new data communication equipments procured by the Federal Government are to conform to Fed Standard 1031 which was adopted from Electronic Industries Association (EIA) standard RS-449 (Ref 9:72). This standard includes both RS-422A and RS-423A electrical specifications and the mechanical and functional characteristics which define the DTE/DCE interface. The Network-to-UNID II interface will be designed to conform to the RS-449 standard, but since many older types of equipments are in use in the AFIT laboratories, the Local-to-UNID II interface will be designed to be RS-232C compatable.

There is no U.S. standard which is equivalent to the electrical, mechanical, and functional characteristics of X.21 (Ref 4:437), but RS-232C and RS-449 are essentially equivalent to the procedural characteristics of X.21bis (Ref 4:437). X.21bis is the analog counterpart to X.21 which is to be used for interfacing analog networks until digital

networks become widely available (Ref 41:238). The appendix of RS-449 contains a mapping of the RS-449 functional circuits with the X.21 functions (Ref 8).

Summary of Requirements

This chapter summarized the requirements for the UNID II and the DELNET. The X.25 access protocol standard was introduced and its correlation to the UNID-II functions was briefly explained. The input requirements (Table I) and the Data Flow Diagrams (Appendix A) of the functional requirements model form the basis for the design and construction of the protoype UNID II described in the following chapters.

III. UNID II Network Module Design and Construction

This chapter describes the design and construction of the UNID II network module. The overall UNID II architecture is based on the block diagram shown previously (Figure 2). This architecture was selected during a preliminary UNID II design project (Ref 14) in which several alternative designs were also considered. The first UNID II configuration considered consisted of an 8086 central processing unit (CPU), two Remote 8089 Input/Output Processors (IOPs), and an area of shared memory for inter-processor communications (Ref 14:66). This configuration was not selected due to the following four reasons:

- It consisted of three hardware subsystems, whereas the UNID II's Data Flow Diagrams (Appendix A) indicated that only two were necessary.
- The added complexity of the CPU interface and arbitration circuitry was not worthwhile, since the 8086 did not provide additional processing capability to the two 8089 subsystems.
- Both 8089's would need to share the single CPU and they would require continual use of the system bus during message processing. Therefore, system bus contention would be unreasonably high.
- The 8089s have limited general-purpose instructions, so this configuration might not have been capable of handling the UNID II's functional requirements.

The second UNID II architecture considered consisted of two hardware subsystems (Ref 14:68). It was similar to the first design iteration, except the two seperate 8089 subsystems had been combined into the same subsystem. This

configuration would have reduced the complexity of the bus arbitration circuitry, but it still had the apparent problems of system bus contention and both IOP's sharing the single CPU.

The final UNID II design (Figure 2) also has two hardware subsystems, but the network I/O and local I/O processing functions are handled seperately. An Intel SBC 86/12A single-board-computer was specified for use as the local module (Ref 14:68). It contains an 8086 CPU and its 32K bytes of random access memory (RAM) can be shared with other processors, via the system bus.

The block diagram for the network module is illustrated in Figure 8. It incorporates an Intel 8086 CPU and an Intel 8089 I/O Processor. Both processors have access to shared system memory, which is accessed over the Multibus, and to resident RAM memory located on the network board. No commercially available microcomputer boards exist with the required 8086/8089 architecture, so the network module had to be developed in-house.

Before the complete network module was constructed, a demonstration network module was designed and wire-wrapped. The demonstration module, shown in Figure 9, is identical to the network module, except that an Intel 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is used to connect the network module to a CRT terminal instead of having the two seperate I/O channels. The USART permits messages to be transferred between the terminal and shared

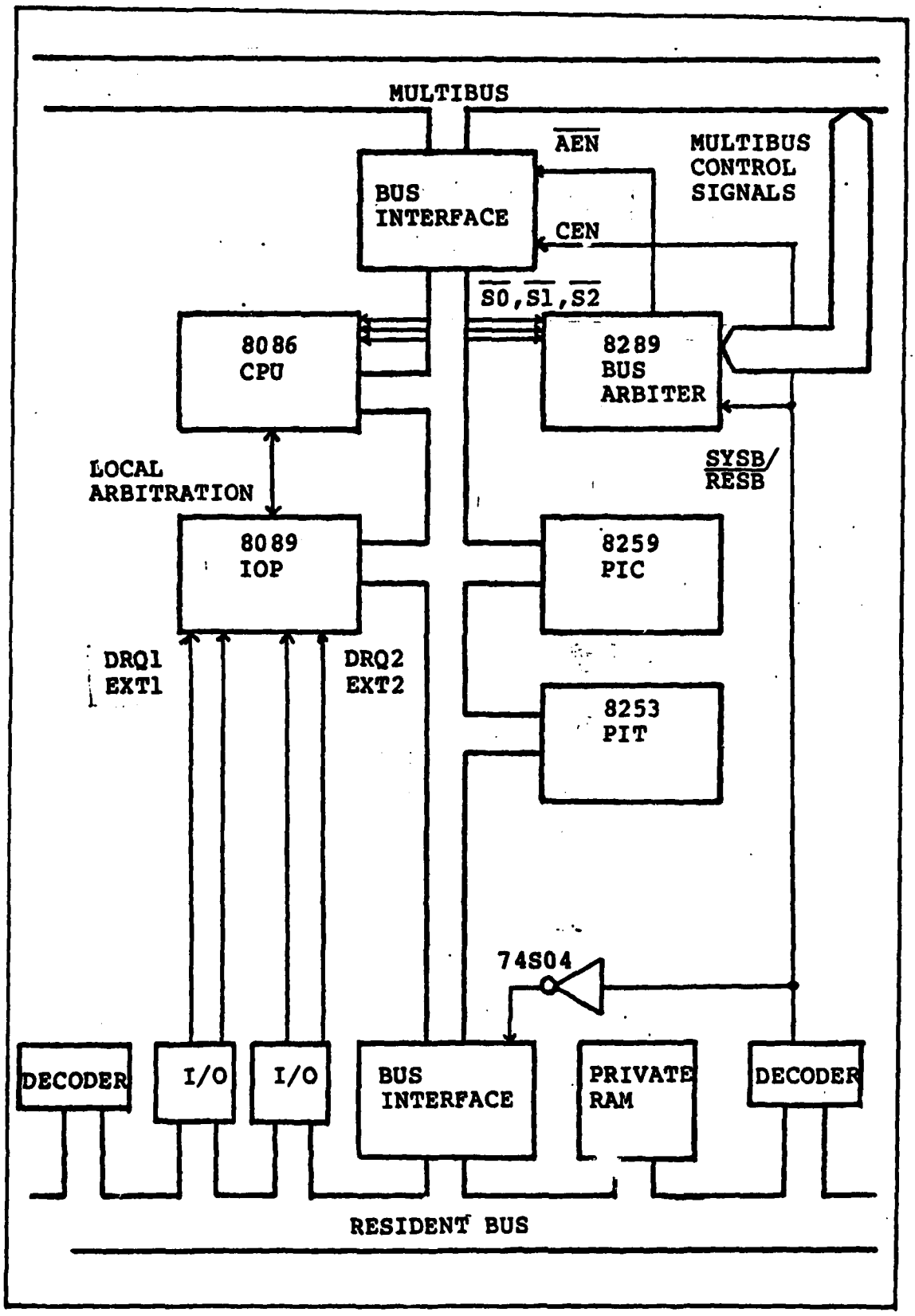


Figure 8. Network Module Block Diagram

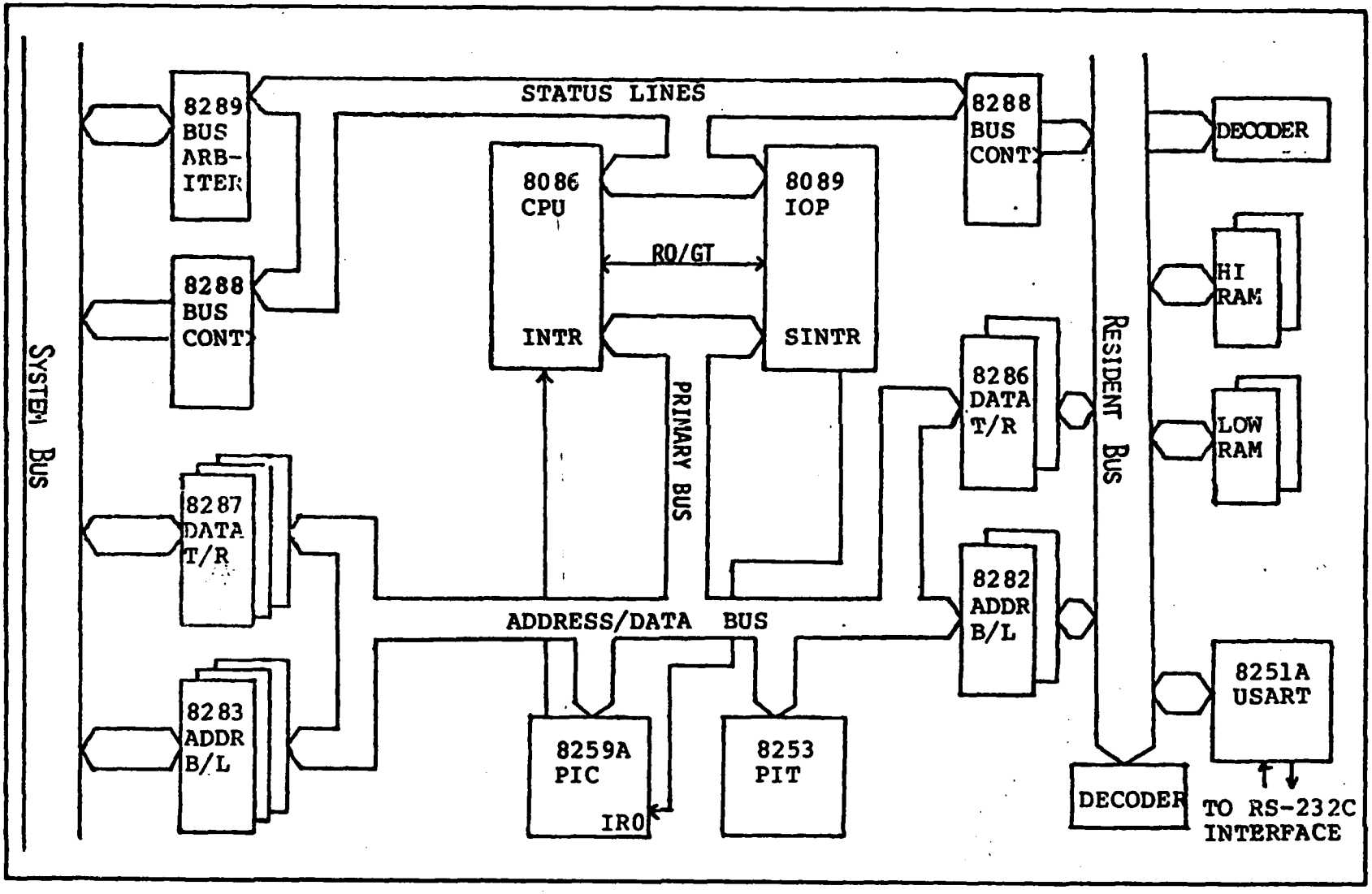


Figure 9. Block Diagram of Demonstration Network Module

memory via the network module. The Intel application note "Prototyping with the 8089 I/O Processor" (Ref 30), was used as a design guide for the construction of the demonstration network module and the software development.

To describe the design philosophy and the operating characterstics of the demonstration network module, the components of the network module will be presented seperately as five operational groups: the 8086 CPU, 8089 IOP, Bus Interfaces, 8289 Bus Arbiter, and the I/O control devices. A description of the construction of the network module is also presented. The physical pin-out diagrams for the major system components are shown in Appendix C. All references made to 80XX and 82XX components refer to Intel Inc. components, unless otherwise noted.

8086 Central Processing Unit (CPU)

The Intel 8086 is a third generation microprocessor. It and the 8088 are identical, except that the 8088 has an 8-bit external data bus, whereas the 8086's external data bus is 16-bits. The 8086 CPU was designed to be assembly language compatable with the 8080A microprocessor (Ref 32:83), and its performance is seven to ten times that of the 2-MHz 8080A (Ref 23:2-3). The 8086 has a much larger application range than the 8080A; it can be utilized in systems requiring only a single processor or in systems with multiprocessor configurations. Also, the 8086 16-bit data bus allows more data to be transferred during each bus cycle as compared to the 8080A.

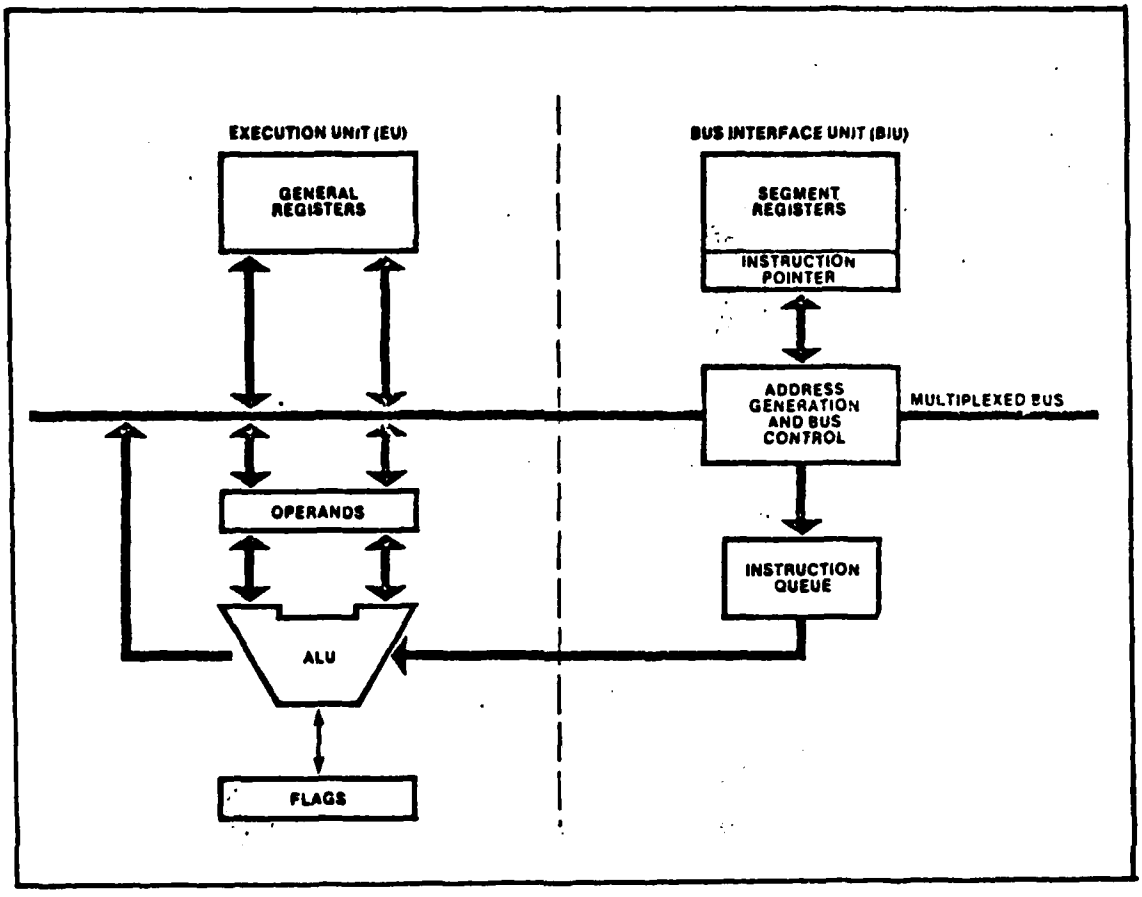


Figure 10. 8086 Execution Unit and Bus Interface Unit

The internal architecture of the 8086 is divided into two seperate processing units, the execution unit (EU) and the Bus Interface Unit (BIU). A block diagram of the EU and BIU is shown in Figure 10. The EU has four 16-bit general registers, two 16-bit pointer registers, and two 16-bit index registers. The EU operates on the instructions from a queue maintained by the BIU. Up to six instructions can be "pre-fetched" by the BIU and placed in the instruction queue. This enables the EU to be more effectively utilized for executing instructions instead of being idle waiting for bus transfers. If an instruction stipulates that a bus

transfer is required, the EU must request assistance from the BIU since the EU has no connection to the external bus (Ref 20:2-5,2-6).

The BIU fetches instructions, reads operands, and writes results. It can access one megabyte (1,048,576 bytes) of memory with its 20-bit address bus. The memory space can also be divided into logical segments of up to 64K bytes each by using the four segment registers: the data segment register, the stack segment register, the code segment register, and the extra segment register. These segment registers serve as an aid to modular software development (Ref 20:2-11).

In order to incorporate a 20 bit address bus and a 16 bit data bus onto a 40 pin chip, Intel assigned several of the 8086 pins dual functions. The lower 16 address lines are time-multiplexed with the 16 data lines (AD0 - AD15) and the other pins with secondary functions are denoted by the signal names in parentheses (Appendix C).

Bus Cycle. To illustrate how the address/data lines are multiplexed, a bus cycle sequence will be explained. All bus cycles consist of at least four clock cycles referred to as "T-states". When performing a Write bus cycle, as shown in Figure 11, the BIU places the 20 bit address of a memory location or an I/O device onto the bus during state T1. Then at state T2, the CPU removes the address from the bus and replaces it with the data to be written. This data will remain on the bus until the completion of the bus cycle at

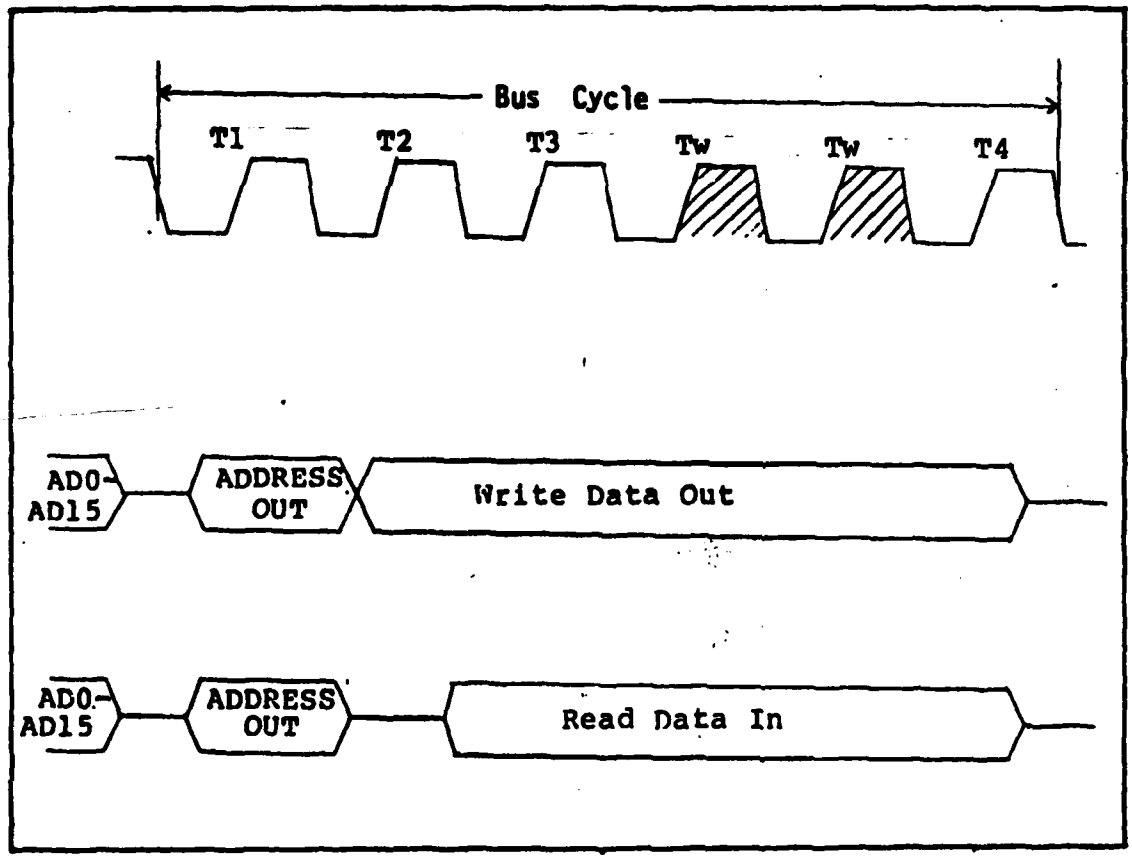


Figure 11. 8086 Address Bus Status During Read and Write Bus Cycles, with Wait States

state T4.

During a Read bus cycle, as shown in Figure 11, the address is again placed on the multiplexed address/data bus during T1, but during T2 the lower 16 address/data lines are three-stated (floated) in preparation for the read cycle. Then during T3, the data is read from the bus and the cycle terminates after T4.

The BIU executes a bus cycle only when requested to by the EU or when it is filling the instruction queue. When the BIU is inactive (not accessing the bus), the clock states are referred to as idle states (TI states). It is

during these idle states that the bus can be utilized by another processor in a multi-processor configuration (Ref 20:4-5 to 4-7).

Ready, Reset, and Clock Generator Circuitry. The clock generator and its associated Ready and Reset circuitry are shown in Figure 12. The clock generator establishes the bus cycle time of the CPU and IOP. A 15-MHz crystal is used for the clock time base and the 8284 divides this frequency by 3 to obtain the 5-MHz clock signal supplied to the CPU and support components. Also, a 2.5-MHz square-wave is supplied to I/O devices from the clock generator's PCLK output.

The clock generator provides the RESET and the READY signals to the 8086 and the 8089. The 8284 has two Ready inputs, RDY1 and RDY2, to permit controlling of two Multi-Master system buses (Ref 20:B-65). Inputs $\overline{\text{AEN1}}$ and $\overline{\text{AEN2}}$ (an overscored signal designation will be used to denote an active low signal) are qualifiers for their respective RDY inputs. If a slow memory or a peripheral device is not ready to accept or to transmit data, it deactivates the RDY1 and RDY2 inputs to the 8284 prior to state T2. The clock generator then deactivates the READY line to the CPU which causes wait states ("TW" states) to be inserted into the bus cycle between T3 and T4 (Figure 11). When the device reactivates the RDY input, the 8284 activates the READY line and the CPU completes the bus cycle at T4.

The READY approach used for the UNID II network module is that of "normally not ready" (Ref 37:8-24). The 8284's

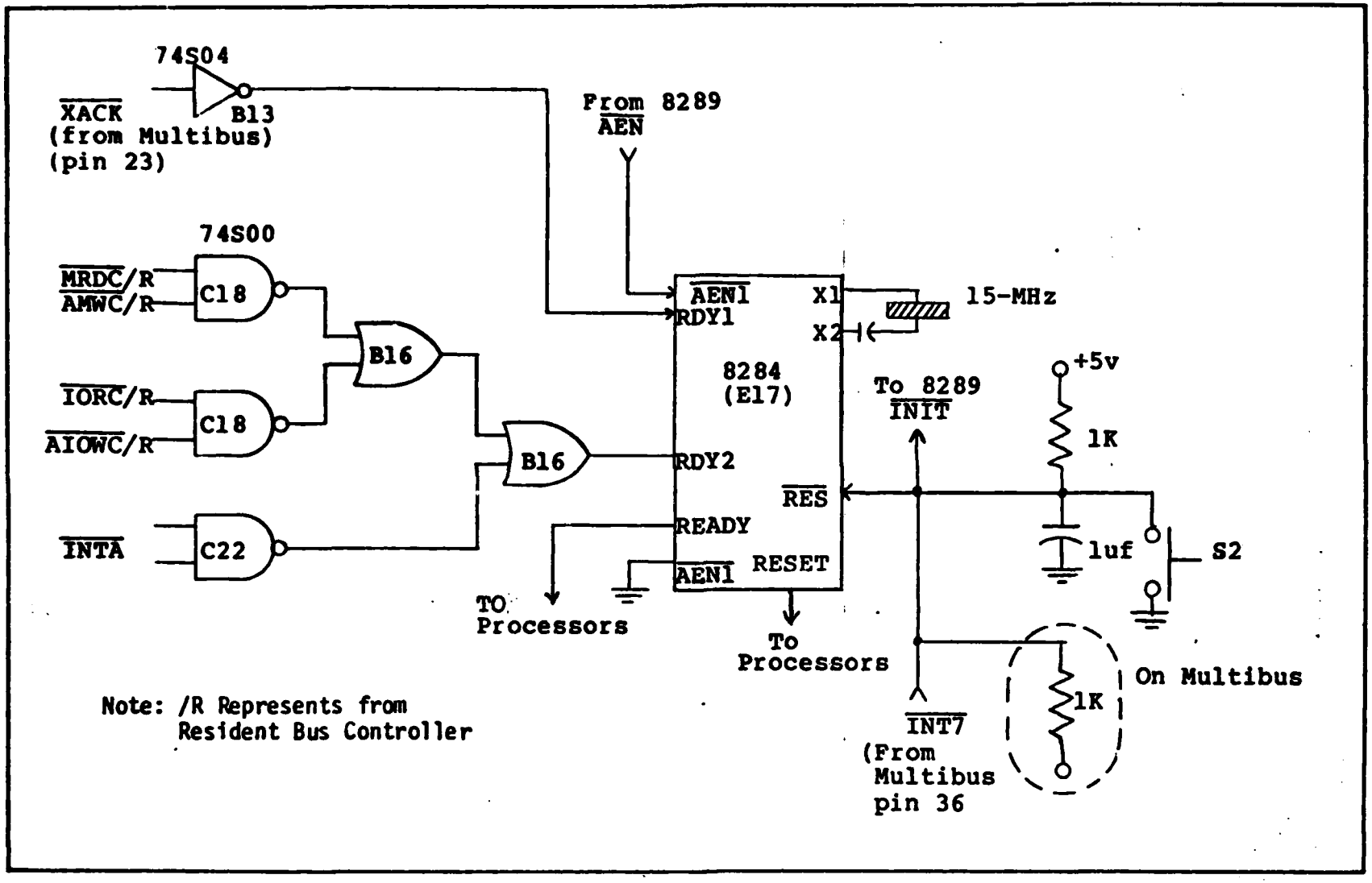


Figure 12. Ready, Reset, and Clock Generator Circuitry

RDY inputs are held inactive until the selected memory or I/O device has had sufficent time to respond to a Read, Write, or Interrupt Acknowledge. To avoid the insertion of a wait state into the bus cycle, the READY signal must not be active (high) within 119ns of the positive transition of the T3 clock cycle. Also, the READY must not change from high to low during the clock low time of T3 and it must satisfy a hold time of 30ns after the T3 positive transition (Ref 37:8-25).

RDY1 input is obtained from the Multibus transfer acknowledge (XACK) signal. It is qualified by the $\overline{\text{AEN}}$ signal from the bus arbiter. The resident bus control signals ($\overline{\text{MRDC}}$, $\overline{\text{AMWC}}$, $\overline{\text{IORC}}$, $\overline{\text{AIOWC}}$, and $\overline{\text{INTA}}$) are applied through the combinational logic (Figure 12) to the RDY2 input. The RDY2 line is disabled (low) until any one of the bus control signal is activated. Only one Multi-Master system bus is used in the network module, so the $\overline{\text{AEN2}}$ input is permanently enabled (tied to ground).

Since the READY is "normally not ready," a program should not assign executable code to the last six bytes of physical memory. The CPU prefetches up to six bytes of instructions and it may try to access non-existent memory. If the READY signal is not enabled when this occurs, the system will enter into an indefinite wait state (Ref 37:8-28).

The network module RESET is obtained from two sources. One is from the push button switch, S2, and the other is

from the local module, via the Multibus $\overline{INT7}$ line (pin 36). The local module can pulse this line to reset the network module.

8086 Modes of Operation. The 8086 CPU can operate in one of two modes, minimum mode or maximum mode. By strapping the $\overline{MN}/\overline{MX}$ pin high (high=1, low=0), the CPU operates in the minimum mode. This mode is used for small, single-processor systems and all bus control signals are obtained directly from the CPU. If the $\overline{MN}/\overline{MX}$ pin is strapped low, the CPU will operate in the maximum mode. In this mode it is necessary that an 8288 Bus Controller and an 8289 Bus Arbiter be used in conjunction with the 8086 CPU to provide bus control functions and for controlling access to the system bus.

The pins which are assigned different functions in the maximum mode are the three status lines ($\overline{S0}$, $\overline{S1}$, and $\overline{S2}$), the request/grant lines ($\overline{RQ}/\overline{GT1}$ and $\overline{RQ}/\overline{GT0}$), the $\overline{LOCK}$ line, and the queue status lines (QS0 and QS1). The three status lines provide the CPU status conditions to the 8288 bus controller and the 8289 bus arbiter. These status bits are decoded by both devices and the appropriate bus control commands are issued by the bus controller. The bus control commands are issued as listed in Table II.

The Reguest/Grant signal lines are designed to be utilized in multiprocessor applications incorporating an 8089 IOP. The request/grant seguence is explained in more detail later in this chapter. The $\overline{LOCK}$ signal line is used

Table II. CPU Status Bit Decoding

Status Inputs			CPU Cycle	8288 Commands
0	0	0	Interrupt Acknowledge	$\overline{INTA}$
0	0	1	Read I/O Port	$\overline{IORC}$
0	1	0	Write I/O Port	$\overline{IOWC}$, $\overline{AIOWC}$
0	1	1	Halt	None
1	0	0	Instruction Fetch	$\overline{MRDC}$
1	0	1	Read Memory	$\overline{MRDC}$
1	1	0	Write Memory	$\overline{MWTC}$, $\overline{AMWC}$
1	1	1	Passive	None

to inform the bus arbiter that the bus is not to be relinguished to another processor until one clock cycle after the execution of the current instruction (Ref 20:4-14). This signal is activated during an interrupt acknowledge seguence or under software control. The queue status lines are used for external monitoring of the CPU's internal instruction queue (Ref 20:4-11 to 4-14).

Interrupt Acknowledgement Sequence. The 8086 has a table of up two 256 interrupt vectors which are stored in RAM locations 000-3FFH. Each vector consists of four bytes; two bytes for the instruction pointer and two bytes for the code segment register, which are used to form the address of the interrupt service routine (Ref 20:A-25). The first 32 interrupt types (type 0-31) at locations 00-7FH are predefined by Intel and they should not be used for other purposes (Ref 20:4-17).

When the CPU receives an interrupt on its INTR line from the 8259A Programmable Interrupt Controller (PIC), it disables any other interrupts and saves the current instruc-

tion register and code segment register contents on the stack. Then the CPU initiates an interrupt acknowledge bus cycle. Since the CPU is operated in the maximum mode, the resident 8288 bus controller decodes the CPU status lines and issues the first of two interrupt acknowledge ($\overline{\text{INTA}}$) signals. This signal informs the PIC that the CPU has acknowledged its interrupt request. When the 8288 issues the second $\overline{\text{INTA}}$, the PIC places a pre-programmed interrupt-vector byte onto the data bus. The CPU multiplies this byte by four to acquire the address of the interrupt vector typ . The CPU program execution is then vectored to the address specified by the pointer values stored in the interrupt vector table at that interrupt type location (Ref 20:A141).

When an interrupt acknowledge sequence is initiated, the CPU activates its $\overline{\text{LOCK}}$ signal to preclude any other bus master from acquiring the bus until the interrupt acknowledgement sequence is completed. The bus controller is operated in its system bus mode, so it also issues a Master Cascade Enable (MCE) signal. However, only one PIC is used by the network module, so this signal is not used (Ref 20:B-77).

8089 Input/Output Processor (IOP)

The 8089 IOP was designed to alleviate the 8086 and 8088 microprocessors from the overhead associated with I/O operations. The 8089 IOP is capable of dynamically translating and comparing data during direct memory access (DMA) and of supporting several terminate conditions (Ref 20:4-

38,39). These DMA terminate conditions are as follows (Ref 34:2-6):

- Byte Count: Terminates when a preset counter decrements to zero
- Mask Compare: Terminates as a match or a mismatch occurs when the transferred data is compared to a fixed pattern
- External Logic: Terminates when an external event activates the 8089's EXT input
- Single Cycle: Terminates after one byte or word is transfered

The first three terminate conditions can be specified seperately or in combinations. The single cycle condition has presedence over the other three terminate conditions.

The address/data lines of the 8089 are identical to those of the 8086. The 8089 can also access one megabyte of memory and 64K bytes of I/O space. Three status lines are provided by the 8089 for decoding by a bus controller and a bus arbiter in the same way as the 8086.

The 8089 has two I/O channels which operate independently. The only interaction between the two I/O channels occurs when both channels attempt to access the bus at the same time. This contention is alleviated by priority assignments. One channel can be given priority over the other or their bus requests can be serviced on alternating bus cycles (Ref 34:2-3,4-5). A channel performing DMA transfers has a higher priority than one performing normal program operations (Ref 34:4-5).

8089 Modes of Operation. The 8089 IOP can be operated in one of two modes, the local mode or the remote mode. In the local mode the 8089 functions as a slave to the 8086 or 8088 CPU that is operating in the maximum mode. The 8089 can also operate as a slave to another 8089 (Ref 34:3-3). The 8089 shares the address latches, data transceivers, and the bus controllers with the CPU. The major shortcomming of this configuration is that only the CPU or the IOP can access the bus at any one time. In the remote mode, the 8089 has its own private local bus. It can access I/O devices on the local bus and execute memory transfers over the shared system bus. This configuration makes the most effective use of the 8089 since it can operate in parallel with other IOPs or CPUs on the system bus (Ref 34:3-3).

The 8089 used in the UNID II network module will be operated in the local mode. As mentioned in the previous thesis report (Ref 14:64 to 68), less system bus contention and a less complex CPU interface are realized when the local mode is used. The 8089 and the network 8086 will share resources over both the system bus and the resident bus.

8086 and 8089 Communications. The 8086 CPU and the 8089 IOP communicate through a block of shared memory. Figure 13 illustrates the memory control structure of the 8089 control blocks. Device initilization begins with the CPU setting up the channel control blocks and the parameter blocks for each of the two IOP channels (Ref 34:4-1). When the 8089 receives a RESET, it must wait for the CPU to

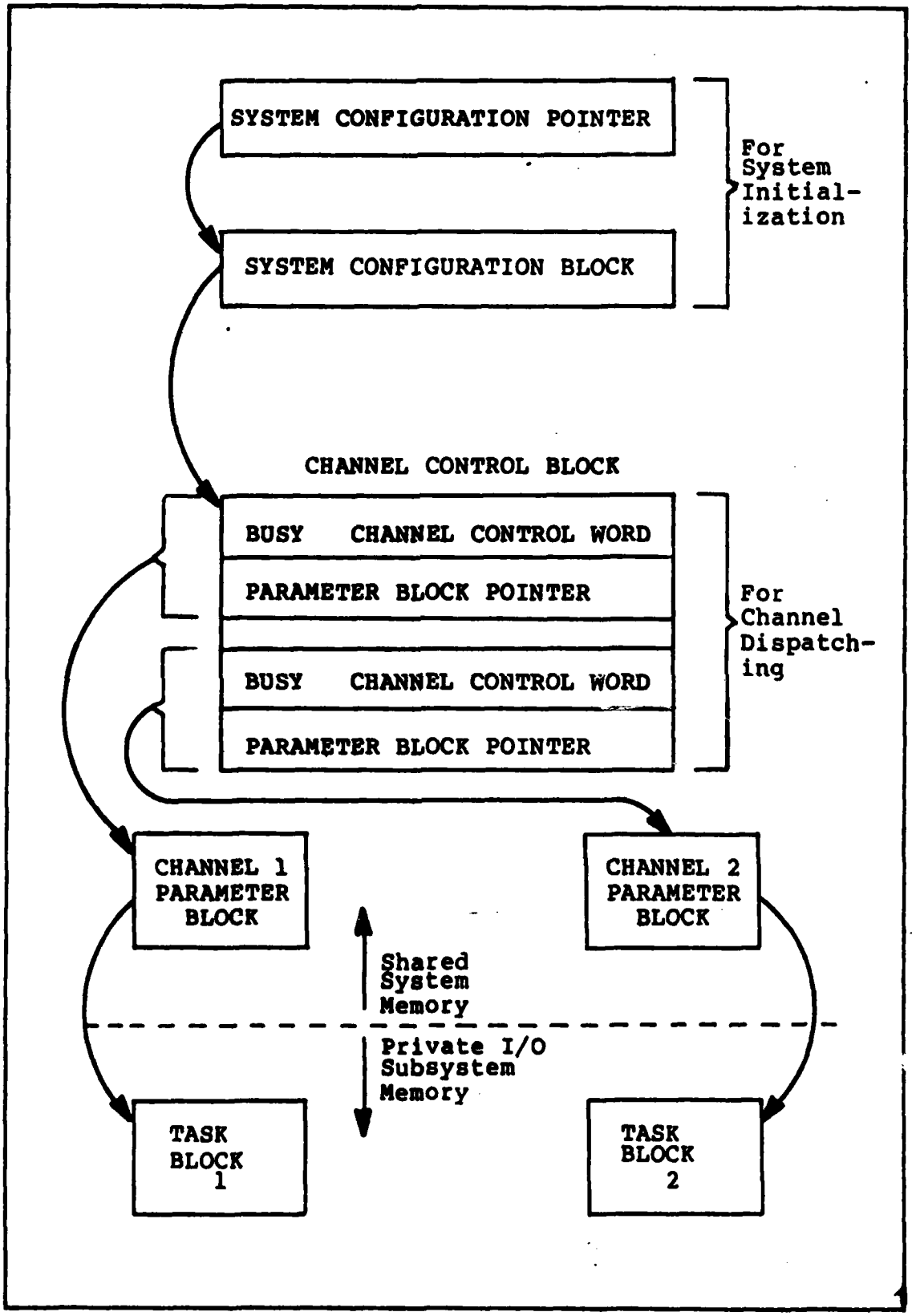


Figure 13. 8086 and 8089 Communication Blocks (Ref 20:3-3)

initiate its initilization sequence by setting its Channel Attention (CA) input high and its channel select (SEL) input to the appropriate state. If the SEL input is low during the first CA after RESET, the 8089 will operate as a master. If SEL is high, it will operate as a slave. For all CA inputs after the first one, the SEL input will function as a select between the two IOP channels (Channel 2 when SEL=1 and channel 1 when SEL=0).

After receiving the RESET and the CA signal, the 8089 begins executing the initilization sequence stored in its internal ROM. The system byte is fetched from system memory at location 0FFFF6H (Ref 20:3-23). The LSB (least significant bit), bit0, determines the physical bus width of the system bus. If bit0=0, an 8-bit system bus is specified and if bit0=1, the system bus is 16 bits. The IOP then reads the system configuration pointer at location 0FFFF8H which directs it to the System Configuration Block (SCB). This block contains the System Operation Command (SOC) byte which informs the IOP the physical I/O bus width and the request/grant mode of operation. The physical I/O bus width is 8 bits if bit0=0 and 16 bits if bit0=1. Request/Grant mode 1 (bit 1=1) is used only when two IOPs are using the same I/O bus (Ref 20:3-35,34:3-2), so for the network module, mode 0 (bit 1=0) will be used. The SCB also contains a pointer to the channel control block. The IOP stores this pointer in an internal register and the location of the channel control block must not be moved after

initilization (Ref 20:3-40).

After its initilization is complete, the IOP clears the channel 1 BUSY flag. The CPU monitors this bit during the 8089's initilization and it takes control again when it detects the BUSY bit clear. After initilization, each time the CPU activates the IOP's CA input, the IOP will set the proper channel BUSY flag (determined by the SEL input) and read the channel control word (CCW) from the channel control block. The CCWs and the parameter blocks must be setup by the CPU prior to issuing any CAs after the first one. This is because the CCWs instruct the IOP as to what type of operation it is to perform (Ref 20:3-41).

Request/Grant Sequence: The request/grant ($\overline{RQ}/\overline{GT}$) line serves as a local arbitrator for the primary bus. It ensures that both the CPU and the IOP do not attempt a bus access at the same time. When the IOP is operated as a slave, it pulses the $\overline{RQ}/\overline{GT}$ line to request the bus from the CPU. When the IOP detects a response on this same line, the IOP takes control of the bus (Ref 20:4-13). Once the IOP has control of the bus the CPU cannot demand it back, it must wait for the IOP to release the bus (Ref 20:4-39). The bus is released in the same sequence as when the bus is requested; the IOP pulses the $\overline{RQ}/\overline{GT}$ line to inform the CPU that it is ready to release the bus.

DMA Transfers. The 8089 IOP can transfer blocks of data between any two address locations; memory-to-memory, memory-to-port, port-to-memory, or port-to-port. Any block

size can be transferred unless the byte count terminate option is specified. In this case the block size cannot exceed 64K bytes (Ref 20:3-27).

The number of bytes transferred during a single DMA cycle depends on the logical bus widths of both the source and destination, and on the address boundary (even or odd address) (Ref 20:4-43). A "WID" instruction (Ref 34:3-7,3-8) specifies the logical bus widths, which may be different from, but not greater than, the physical bus widths.

The DMA transfer can have three modes of synchronization (Ref 20:3-29): unsynchronized, source synchronization, or destination synchronization. Unsynchronized transfers are typically used in memory-to-memory transfers. The source synchronization is typically used when the source is an I/O device and the destination is a memory location. Likewise, destination synchronization is used most often when the source is memory and the destination is an I/O device. Onlyone type of synchronization may be specified. During a DMA transfer, memory addresses are incremented by one if a byte is transferred, and by two if a word is transferred. I/O ports are not changed (Ref 34:4-10).

When using source or destination synchronization, the I/O control device must initiate the DMA transfer by activating the IOP's DRQ inputs (DRQ1 for channel 1 and DRQ2 for channel 2). All DMA transfers pass through the IOP to allow for optional mask/compare or translate operations. Each transfer consists of at least two bus cycles; a fetch of

the data from the source into the IOP, and a store of the data from the IOP to the destination. The IOP may need to assemble or disassemble the data depending on the source and destination logical bus widths (Ref 20:4-48).

The DMA transfer cycle is terminated when one of the four terminate conditions occurs. If more than one terminate condition is specified, a displacement value is added to the IOP's Task Pointer register to indicate where normal program execution is to resume. The displacement can be 0, 4, or 8 bytes. If two or more terminate conditions occur simultaneously, the largest specified displacement is used (Ref 34:4-11).

The fastest possible DMA transfer rate occurs for 16-bit unsynchronized transfers (not memory-to-memory) and for source or destination synchronized transfers. For these cases, two bytes of data are transferred every 8 clock cycles. With a 5-MHz clock, this results in a DMA transfer rate of 1.25 million bytes per second (Ref 34:4-13). If performing memory-to-memory transfers, an extra three clock cycles are required per transfer. Mask/Compare operations also require an extra three clock cycles and if the translate option is specified, seven extra clock cycles are required per transfer (Ref 34:4-14).

Bus Interface

The bus interface logic is shown in Figure 14. The bus interface between the system bus and the resident bus are identical, except the system bus uses inverting address

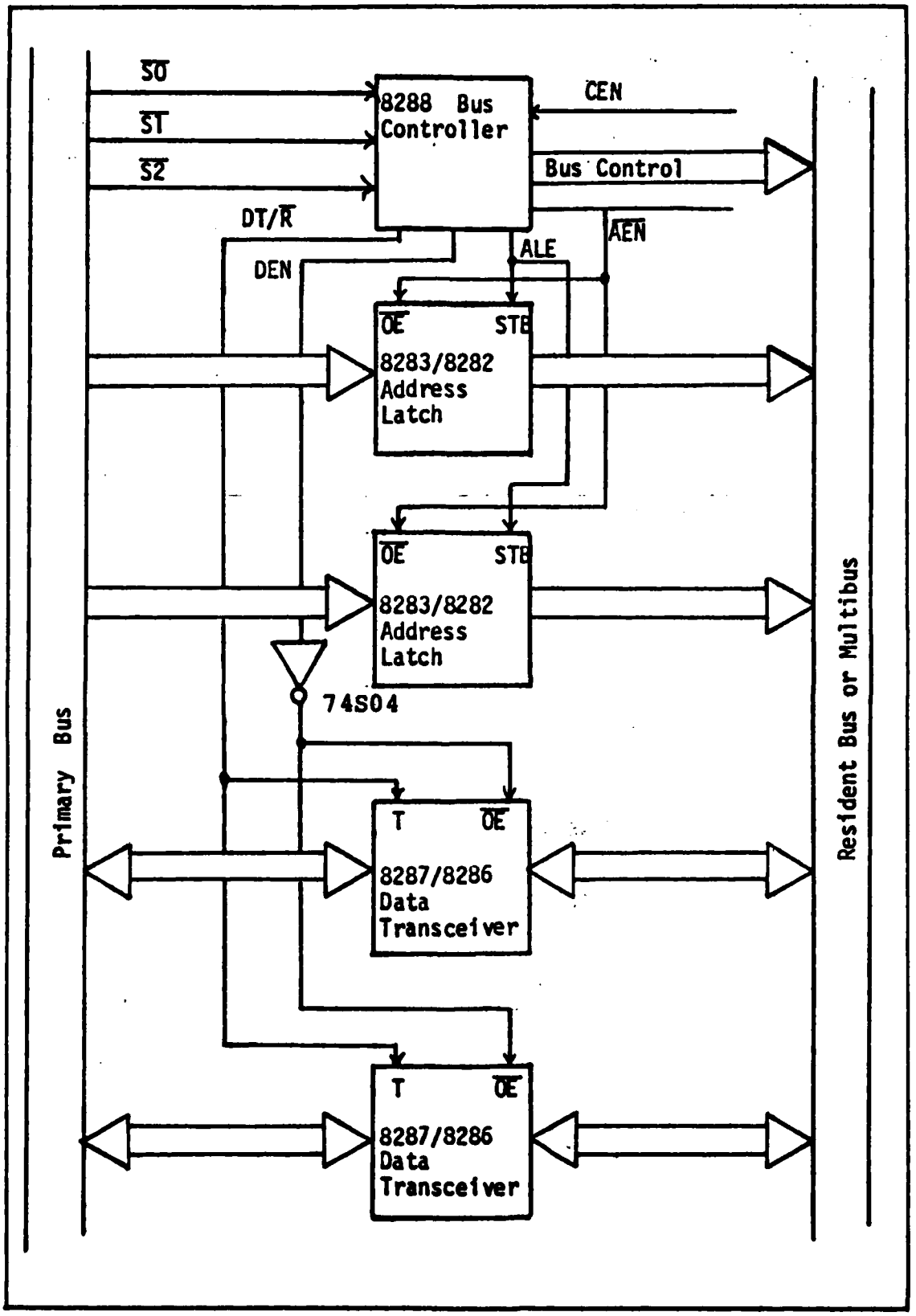


Figure 14. Bus Interface Block Diagram

latches and data transceivers to provide compatability with the Intel Multibus. The interface to the Multibus must be capable of communications with both 8 and 16-bit processors, therefore, three data transceivers are required for interface to the Multibus (Ref 37:9-12). The outputs of the resident bus' address latches are permanently enabled so address status is continuously available to the address decode logic. Only two data transceivers are necessary for interfacing to the resident bus.

The heart of the bus interface is the 8288 Bus Controller. It provides the bus command and control signals when the CPU is being operated in the maximum mode. The bus controller decodes the CPU or IOP status signals as mentioned previously (Table II).

A strapping option allows the bus controller to operate in one of two modes. If the IOB pin is strapped low, the bus controller operates in the system bus mode. In this mode the controller waits for arbitration logic to inform it when the bus is free for use. This is indicated by activation of its Address Enable ($\overline{AEN}$) line by the 8289 bus arbiter. The bus data tranceivers are controlled by the Data Enable (DEN) and Data Transmit/Receive (DT/$\overline{R}$) lines. If the IOB pin is strapped high, the bus controller operates in the I/O bus mode and its I/O command signals are not affected by the $\overline{AEN}$ input. The I/O bus tranceivers are then controlled by the bus controller's $\overline{PDEN}$ and DT/$\overline{R}$ command outputs. When memory is being accessed in this mode, the memory command

outputs ($\overline{\text{MRDC}}$, $\overline{\text{MWTC}}$, and $\overline{\text{AMWC}}$) must wait for the $\overline{\text{AEN}}$ line to be activated.

Both 8288 bus controllers on the network module will be operated in the system bus mode. The system bus controller is operated in this mode because it must wait for its AEN line to be activated by the bus arbiter, indicating that the system bus is available. The resident bus controller is operated in the system mode because it must provide access to both memory and I/O devices connected to the resident bus. As previously shown in Figure 8, their Command Enable (CEN) inputs are controlled by address decode logic. The command outputs of the bus controllers will be disabled when their respective CEN inputs are low. The bus controller issues an Address Latch Enable (ALE) signal to the address latches at the begining of each machine cycle. This strobe signal latches the current status of the primary bus' address/data lines into both sets of address latches.

Data Transceiver Enable Logic. The enable logic for the system bus and resident bus data transceivers is shown in Figure 15. When an even addressed data byte is to be transferred over the system bus the data transceivers, A17 and A18 are enabled. The data is then transferred over the lower 8 data lines of the Multibus. If an odd addressed data byte is to be transferred, data transceiver A16 is enabled, A17 and A18 are disabled, and the data is again transferred over the lower 8 Multibus data lines. Even and odd addressed data words are transferred over the entire 16

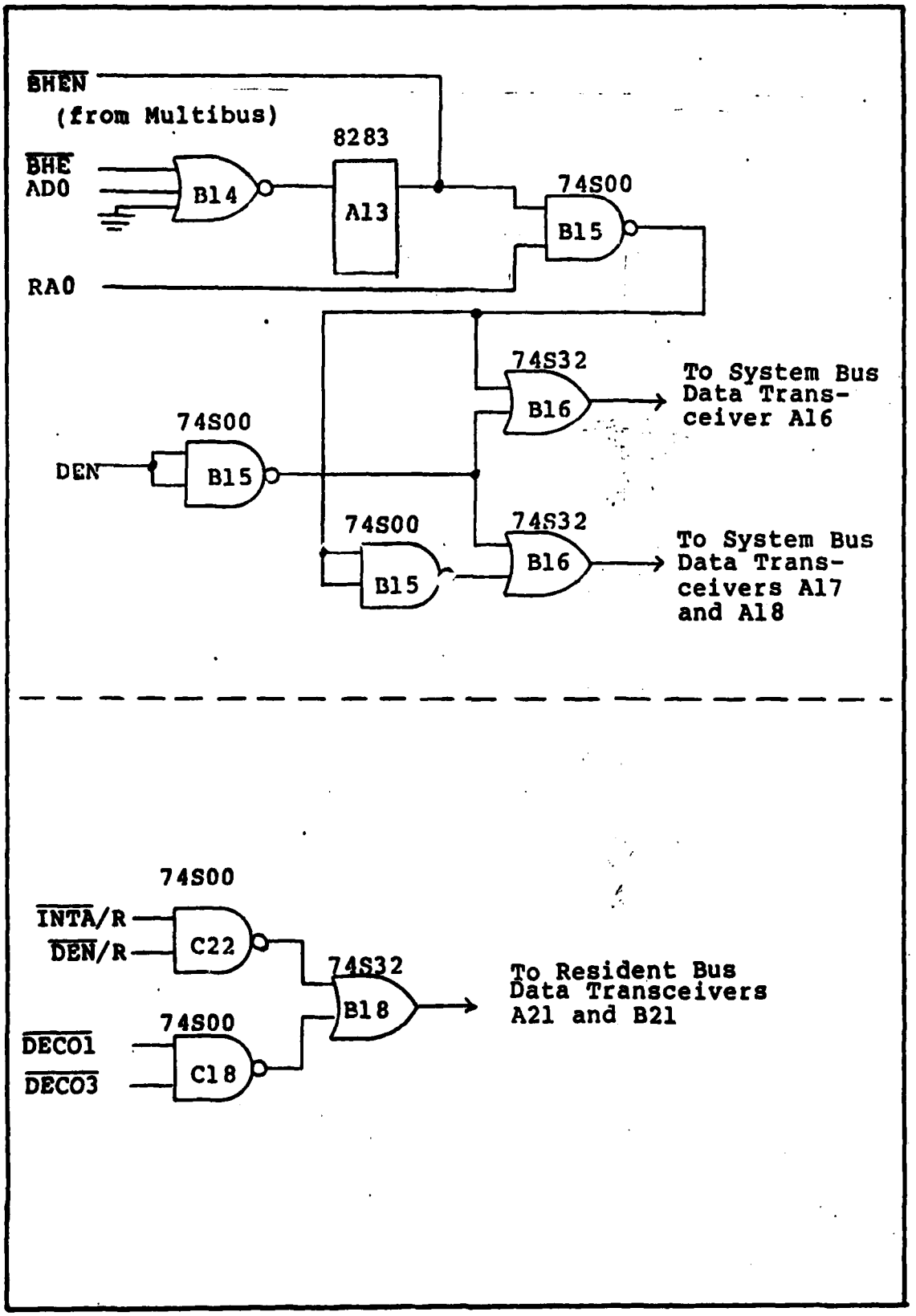


Figure 15. Data Transceiver Enable Logic

data lines by enabling A17 and A18.

The output enable ($\overline{OE}$) signal for the resident bus data transceivers is obtained from the inverted DEN output of the resident bus controller. The $\overline{OE}$ signals could have been obtained from the MCE/$\overline{PDEN}$ line of the bus controller, but it was discovered during testing (Chapter IV) that the data transceivers have to be disabled during accesses to the 8253 Programmable Interval Timer (PIT) and 8259A PIC. The enable logic shown in Figure 15 was designed to handle this situation as well as the situation when an interrupt acknowledge bus cycle is being performed. When an $\overline{INTA}$, $\overline{DECO1}$ (PIT select), and $\overline{DECO3}$ (PIC select) are all disabled (high), the resident bus data transceivers respond to the DEN output of the bus controller. When DEN=1, the data transceivers are enabled. If any one of the $\overline{INTA}$, $\overline{DECO1}$, or $\overline{DECO3}$ lines is enabled, the data transceivers will be disabled.

8289 Bus Arbiter

The Intel 8289 bus arbiter, in conjunction with the 8288 bus controller, permits an 8086, 8088,or an 8089 processor to be interfaced onto a multi-master system bus. The bus arbiter handles the synchronization of bus transfers over the system bus and it provides the arbitration necessary to ensure that only one bus master has control of the bus at any one time (Ref 20:A-113).

The processor (CPU or IOP) is not aware of the presence of the bus arbiter. The processor functions as if it had

Table III. 8289 System Bus Request/Surrender Conditions as a Function of Processor Status

		Modes and Mode Signal Levels						
		System and Local Buses		System Bus Only	System and Loc Mem Bus Loc I/O		System Mem and Local I/O Bus	
Status lines from 8086,8088 or 8089: S2 S1 S0		1	1	1	0	0	0	$\overline{IOB}$
		1	1	0	1	1	0	RESB
		1	0	X	1	0	X	SYSB/ $\overline{RESB}$
I/O Commands	0 0 0	*	#	*	#	#	#	
	0 0 1	*	#	*	#	#	#	
	0 1 0	*	#	*	#	#	#	
Halt	0 1 1	#	#	#	#	#	#	
Memory Commands	1 0 0	*	#	*	*	#	*	
	1 0 1	*	#	*	*	#	*	
	1 1 0	*	#	*	*	#	*	
Idle	1 1 1	#	#	#	#	#	#	

access to the system bus at all times. If the processor does not have control of the system bus and if the bus is unavailable, the bus arbiter will prevent the system bus controller, the data transceivers, and the address latches from accessing the system bus by holding the Address Enable ($\overline{AEN}$) line high. An acknowledgement signal will not be received by the processor until the bus transfer is complete, so if the bus is unavailable, the processor is entered into wait states (Ref 20:B-82).

Strapping options permit the bus arbiter to operate in a combination of two basic modes: the I/O peripheral bus mode and the system bus mode. Table III lists system bus access and surrender conditions for the different config-

urations obtained by strapping the bus arbiter's IOB and RESB pins high or low. The system bus/resident bus (SYSB/$\overline{\text{RESB}}$) input does not have any affect on the bus arbiter when the RESB pin is strapped low (indicated by an X). The "*" represents when the multi-master system bus is requested. The "#" represents when the multi-master system bus can be surrendered (Ref 20:B-84,34:6-6).

The network module has both a system bus and a resident bus, so the bus arbiter is configured to operate in the Resident bus mode ($\overline{\text{IOB}}$ and RESB both high). In this configuration, as shown in Table III, the system bus can be surrendered anytime while the resident bus is being used (SYSB/$\overline{\text{RESB}}$=0), or during a processor "halt" or "idle" state.

Bus Select Logic. The control signal connections between the bus arbiter and the bus controller are shown in Figure 16. When the CPU or the IOP begins a bus cycle, it places a memory or device address onto the primary bus. The status signals are then decoded by the bus arbiter and the two bus controllers. Both bus controllers then issue an ALE signal to their respective address latches, which latches the current state of the address/data lines.

The comparators, C14 and C15, compare the S1 switch settings to the upper four address lines (RA16-RA19). If the address is a system memory address, the SYSB/$\overline{\text{RESB}}$ input of the bus arbiter goes high, causing the bus arbiter to initiate a system bus request and the Command Enable (CEN) input to the resident bus controller goes low, disabling all

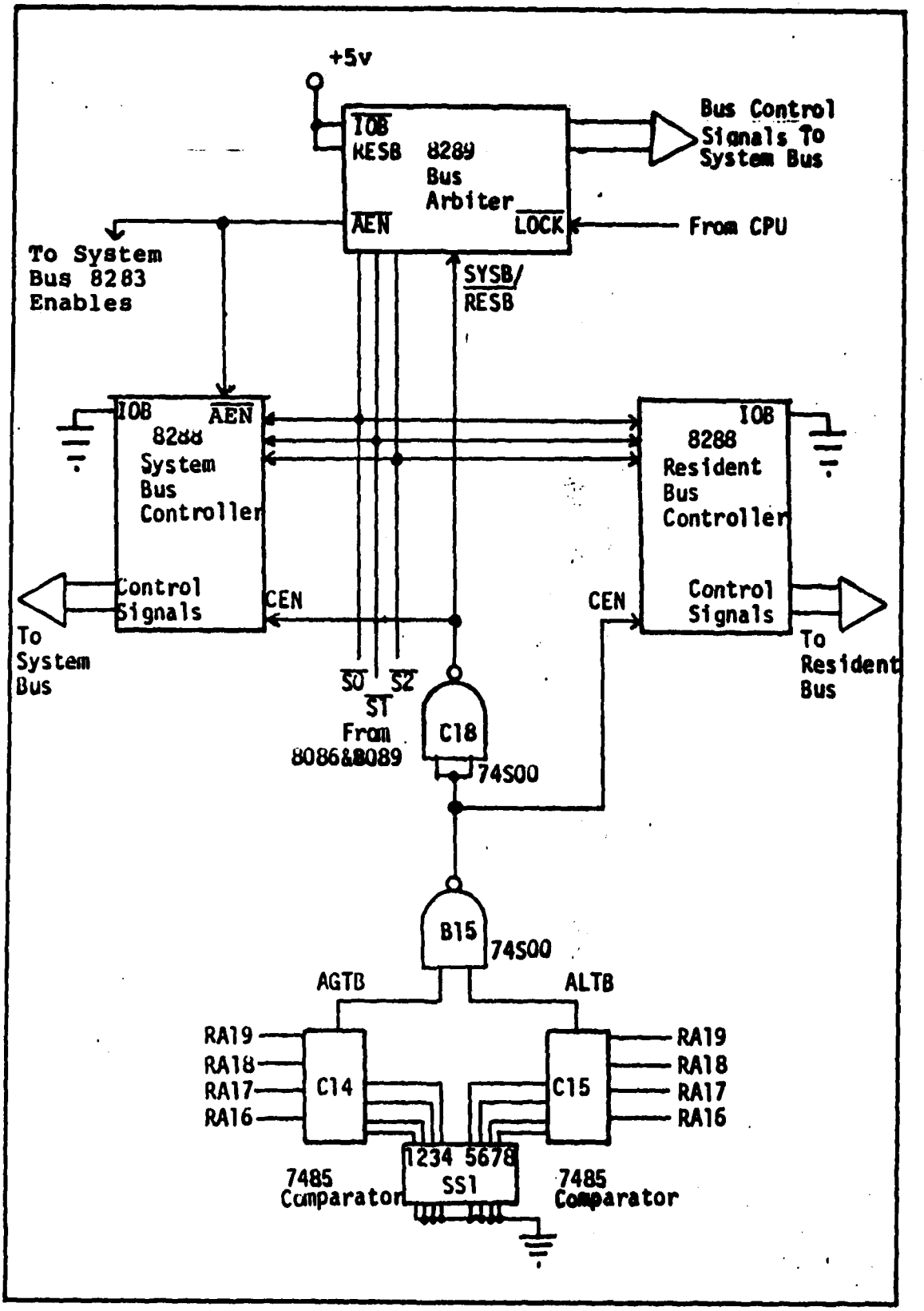


Figure 16. Bus Select Logic

of its control outputs. The CEN input to the system bus controller goes high placing it in an enabled state, but it must wait for an Address Enable ($\overline{AEN}$) strobe from the bus arbiter to inform it when the bus is free. When the $\overline{AEN}$ line is activated by the bus arbiter, the system bus controller issues the commands corresponding to the instructions decoded from the processor status lines (Table II).

If the address placed on the primary bus was for a resident memory location or a peripheral device, the system bus controller would be disabled and the SYSB/$\overline{RESB}$ input to the bus arbiter would be low. As shown in Table III, this would inform the bus arbiter that the system bus could be surrendered to another bus master. The CEN input to the resident bus controller would be high, enabling all of its control outputs and permitting it to complete the bus cycle. The resident bus controller's $\overline{AEN}$ line is permanently enabled (tied to ground), so it does not have to wait for an enable from the bus arbiter to issue commands.

System Bus Surrender/Request Sequence. As mentioned previously, the 8289 bus arbiter is connected in the Resident Bus mode ($\overline{IOB}$=1, RESB=1). In this configuration, the state of the SYSB/$\overline{RESB}$ input informs the bus arbiter whether the system bus is to be requested or if it can be surrendered (Ref 20:B-87). The bus arbiter is also connected in a serial resolving priority configuration, asshown in Figure 17 (Ref 20:A-115). The network bus arbiter has the higher priority for the system bus over the local module bus

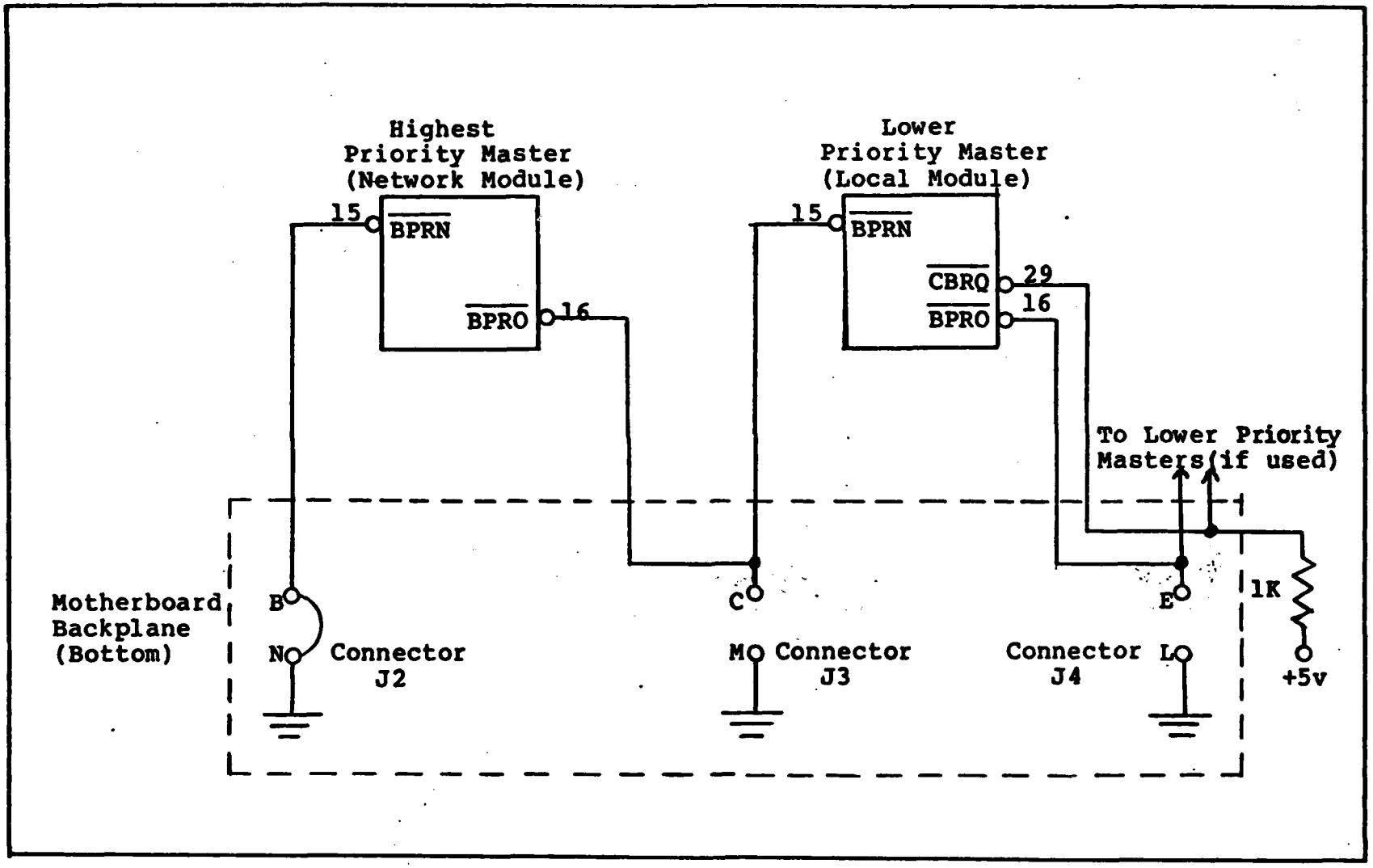


Figure 17. Serial Priority Resolution (Ref 19:2-23)

arbiter. This priority scheme is established by shorting the Bus Priority Out ($\overline{\text{BPRO}}$) line of the network bus arbiter to the Bus Priority In ($\overline{\text{BPRN}}$) input of the local bus arbiter. This shorting connection is located on the Multibus motherboard.

If the network bus arbiter does not have control of the system bus when the CPU or IOP wants to access system memory, it must request the bus from the local bus arbiter. This is performed by issuing a bus request on its bidirectional Common Bus Request ($\overline{\text{CBRQ}}$) line and by driving its $\overline{\text{BPRO}}$ line high. The local bus arbiter will then release the bus after its current bus cycle unless its $\overline{\text{LOCK}}$ input has been set.

When the network bus arbiter acquires the system bus it sets its $\overline{\text{BUSY}}$ line high to indicate that the bus is being used. It also sends the $\overline{\text{AEN}}$ signal to the system bus controllers, 8284 clock generator, and output enables of the system bus address latches. The $\overline{\text{AEN}}$ signal sent to the clock generator's $\overline{\text{AEN1}}$ input is used to qualify its RDY1 input. If the bus arbiter had not been able to acquire the bus in sufficent time to complete the processor's bus cycle, the clock generator would use the status of its $\overline{\text{AEN1}}$ and RDY1 inputs to insert wait states (via the READY line) into the bus cycle (Ref 20:A-23 to A-25).

Once the network bus arbiter has acquired the system bus it will hold it until the bus is requested by another bus master during one of the surrender conditions (Table

III). The bus arbiter will not voluntarily surrender the system bus. If another processor does not request the bus, the bus arbiter will hold onto it. This eliminates the need for the bus arbiter to initiate a bus request each time the system bus is to be accessed (Ref 20:A-122). If the processor had activated the bus arbiter's $\overline{\text{LOCK}}$ input, the bus would not be released until the LOCK was deactivated (Ref 34:6-6).

I/O Control Devices

The Input/Output control devices used in the network module are the 8251A USART, 8253 Programmable Interval Timer (PIT), and 8259A Programmable Interrupt Controller (PIC). The 8251A USART (Ref 20:B-131) is connected to an RS-232C connector through a Motorola MC1488 line driver and MC1489 line receiver. A 2.5 MHz clock signal is provided to the USART by the 8284 clock generator's PCLK output. The 8253 Programmable Interval Timer (PIT) is programmed to provide the square-wave baud rate signal required by the USART (Ref 21:10-159 to 10-169). The 8259A PIC is used to support the vectored interrupt capabilities of the 8086 (Ref 20:B-106 to B-123).

Device Decode Logic. Figure 18 shows the control signal connections between the I/O devices and RAM connected to the resident bus. A 74S288 PROM (Programmable Read Only Memory) (Ref 42:182-189) provides the address decode logic for selecting the peripheral I/O control devices and the resident RAM memory. The memory locations of the RAM memory

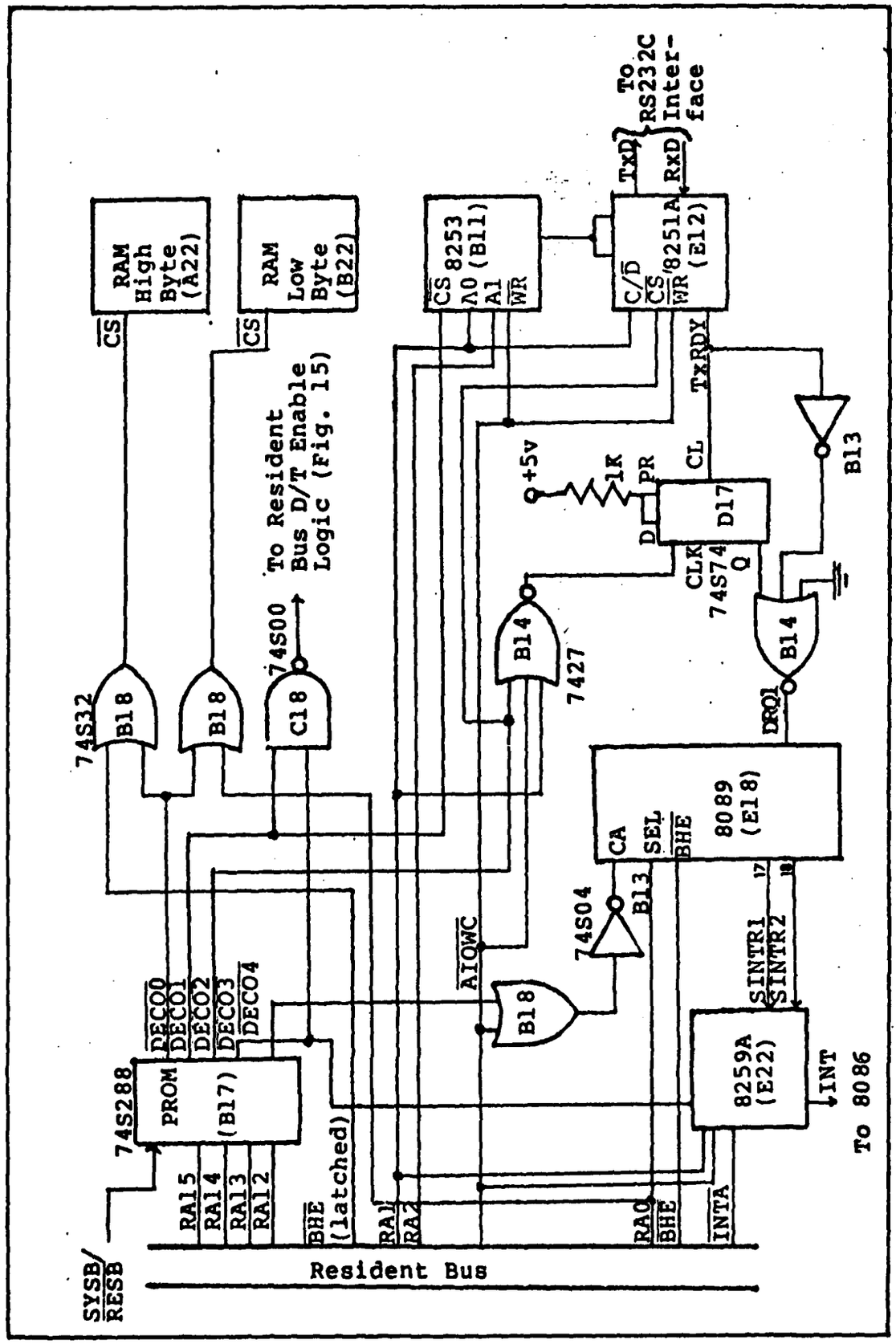


Figure 18. Device Decode Logic

Table IV. Resident RAM and I/O Addressing

PROM Decode	Device	Memory Address (Hex)	Total Memory Space
$\overline{\text{DECO0}}$	RAM Memory	0000 - 3FFF	0000 - 3FFF
$\overline{\text{DECO1}}$	5253 PIT	4000 - 4006	4000 - 4FFF
$\overline{\text{DECO2}}$	8251A USART	5000 - 5002	5000 - 5FFF
$\overline{\text{DECO3}}$	8259A PIC	6000 - 6002	6000 - 6FFF
$\overline{\text{DECO4}}$	8089 IOP	7000 - 7001	7000 - 7FFF

and the port addresses of the I/O control devices are listed in Table IV. The total memory space column lists the address range in which the respective decode signals are active.

Two 8K x 8, EDH8808 (Ref 8) static RAM chips are utilized for the resident memory. One chip serves as the storage location for high-byte data (data lines RD8-RD15). The other chip holds the low data byte (RD0-RD7). The total resident memory space is an 8K X 16 block with 20-bit memory addresses 00000H to 3FFFFH. The lower 1K bytes (addresses 00000 - 3FFH) should be reserved for the 8086 vectored interrupt tables (Ref 20:2-25).

Since two 8-bit RAM chips are connected to the 16-bit resident bus, provisions had to be made to allow for both even and odd, byte and word accesses. The RA0 address line was logically ORed (B18) with PROM output $\overline{\text{DECO0}}$ for selecting the low byte of memory on an even byte or word access. The $\overline{\text{BHE}}$ signal from the CPU and IOP was also ORed (B18) with the $\overline{\text{DECO0}}$ line to select the high byte of RAM for odd byte or odd word accesses. Since RA0 is used as a condition for

chip select, the address/data connections to the resident RAM are RA1-RA13.

The three I/O devices, the 8251A USART, 8253 PIT, and 8259A PIC are 8-bit devices. Therefore, the RA0 signal was used as a condition for their chip selects ($\overline{CS}$), and they were all located at even port addresses. The PIT and the USART could have been positioned on odd port addresses, but the PIC must have an even port address (Ref 20:A-140). The data ports of the PIC and PIT connect directly to the primary bus.

The 8089 IOP is selected whenever the CPU initiates a "write" to port 7000H or 7001H. The IOP's channel attention (CA) input is then activated whenever the PROM $\overline{DECO4}$ output and the resident bus controller's $\overline{AIOWC}$ signal are both low. The RA0 signal determines the state of the IOP's SEL input.

DMA Control Logic. When performing DMA transfers between memory and an I/O port, the I/O device typically provides the source/destination synchronization to the IOP (Ref 20:4-48). For the demonstration network module testing (described in the next chapter), the DMA transfer of data from a shared memory location to an I/O port (the USART) will be performed. Therefore, destination synchronization (Ref 20:3-29) will be used.

The DMA destination synchronization is achieved by controlling the IOP's DMA request input (DRQ1) with the USART's TxRDY output signal (Ref 30:12). The TxRDY line will go high whenever the USART is ready to accept transmit

data (Ref 22:8-37). The TxRDY line is connected to the CLEAR (CLR) input of a D-type flip-flop (F-F) (D17) (Figure 18). When the CLR input is high, the F-F's Q output will toggle on a positive transition of its CLK input (Ref 43:5-22). The F-F's D and PR (preset) inputs are tied high, so its Q output will toggle from 0>1 when its CLR input is high and a positive transition occurs on its CLK input. The F-F's CLK input is obtained from the output of a 3-input NOR gate (B14). Two of the NOR gate's inputs (RA0 and $\overline{\text{DECO2}}$) will be low when the USART is selected. The other input ($\overline{\text{AIOWC}}$) will be activated when the I/O issues a Write command. When the $\overline{\text{AIOWC}}$ is activated "AND" the USART is selected, the CLK signal will transition from 0>1.

The destination synchronized DMA transfer will be controlled as follows: When the DMA transfer is initiated (XFER instruction), the IOP will fetch the data (16-bit words) from its source address (memory location). The IOP must then wait for the DRQ input to be high before it can complete the transfer by sending the data to the destination (USART data port). Initially the TxRDY line will be high (ready to accept transmitter data) and the Q output will be low (Figure 18). Since the inverted TxRDY line and the Q output are inputs to the NOR gate supplying the DRQ1 signal, the DRQ1 will be high and the first data byte will be transferred to the USART. When the $\overline{\text{AIOWC}}$ signal is activated, indicating a "write", the F-F's CLK input will transition from 0>1, causing its Q output to go high. This,

in turn, causes the DRQ1 input to be low during the T3 positive transition of the IOP's "store" bus cycle.

The TxRDY line will go low when the data is loaded into the USART's transmit buffer and the F-F's Q output will be reset. However, since the Q and inverted TxRDY signals are "NOR" conditions for DRQ1, the DRQ1 input will remain low until the data is transmitted by the USART and the TxRDY signal again goes high. If the DRQ1 input does not go high before the positive transition of T4, "idle' clock cycles will be inserted into the IOP's bus cycle before the next "store" cycle is initiated (Ref 20:4-49). When TxRDY goes high, DRQ1 will also go high informing the IOP that another data byte can be transferred. The IOP will transfer the second byte and then fetch the next word to be transferred. The DMA cycle then repeats until a terminate condition is satisfied.

Network Input/Output Ports

Two Signetics 2652-1 Multi-Protocol Communication Controlers (MPCCs) (Ref 18:819-836) will be utilized as the I/O interfaces between the network module and the computer network. These devices, being "multi-protocol," will enhance the UNID II's ability to meet its flexibility requirement. The MPCCs support the bit-oriented (BOP) and byte control (BCP) data link control protocols listed in Table V.

The MPCCs transmit and receive serial data in the frame format presented previously (Figure 7). They automatically

Table V. Data Link Protocols Supported by MPCCs

	Company or Standards Organization	Protocol
Bit Oriented Protocols	IBM	Synchronous Data Link Control (SDLC)
	ANSI	Advanced Data Communication Control Procedure (ADCCP
	ISO	High-Level Data Control (HDLC)
Byte Control Protocols	DEC	Digital Data Communications Message Control (DDCMP)
	IBM	Binary Synchronous Communications (BISYNC) (external CRC)

perform zero insertion and deletion to prevent any bit sequence in the frame to be misinterpreted as a flag, an abort, or an idle. After the beginning flag transmission, a 0 is inserted into the serial data after five successive 1s have been transmitted. Also, when receiving, the number of successive 1s are counted. If five successive 1s are received, the sixth bit is deleted if it is a zero (Ref 45:109).

The 2652s accomplish error checking by performing a cyclic redundancy check (CRC) over the entire frame (between flags) for bit-oriented protocols, and over only the information field for byte control protocols. The inverted remainder of the CRC (CCITT-CRC is used) is transmitted/

received in the 16-bit Frame Sequence Field (Figure 7) for BOP. If BCP is being used, the error information is transmitted/received as the two successive characters following the last data character (Ref 18:821).

The control signal connections to the MPCC devices are shown in Figure 19. Only one 2652 is shown, since the connections to the other one is identical, except that a different chip enable (CE) signal will be used. The address decoder PROM (B17) will supply the CE signals. Address lines RA0 - RA2 are used to select the MPCC's internal registers. The BYTE signal determines the data bus transfer width; when BYTE =1, 8-bits are transfered, and when BYTE = 0, 16 bits are transferred. The BYTE pin is connected directly to the processor's bus high enable ($\overline{BHE}$) line. The Read/Write ($\overline{R}/W$) signal controls the direction of the data bus transfer. This signal is active low for "read" operations and high for "writes".

The activation of the Data Bus Enable (DBEN) signal is more complicated to achieve than are the other control signals. This signal should not be strobed until at least 50 nanoseconds (ns) after the CE, A2-A0, BYTE, and $\overline{R}/W$ controls have been set-up (Ref 18:830). To achieve this, the control logic (Figure 19) was designed to enable DBEN on the positive transition of DEN during a "read" bus cycle, and on a negative transition of the IOWC during a "write" bus cycle. The CE, A2-A0, BYTE, and $\overline{R}/W$ are all set-up prior to, or during, the low clock period of state T2 of the

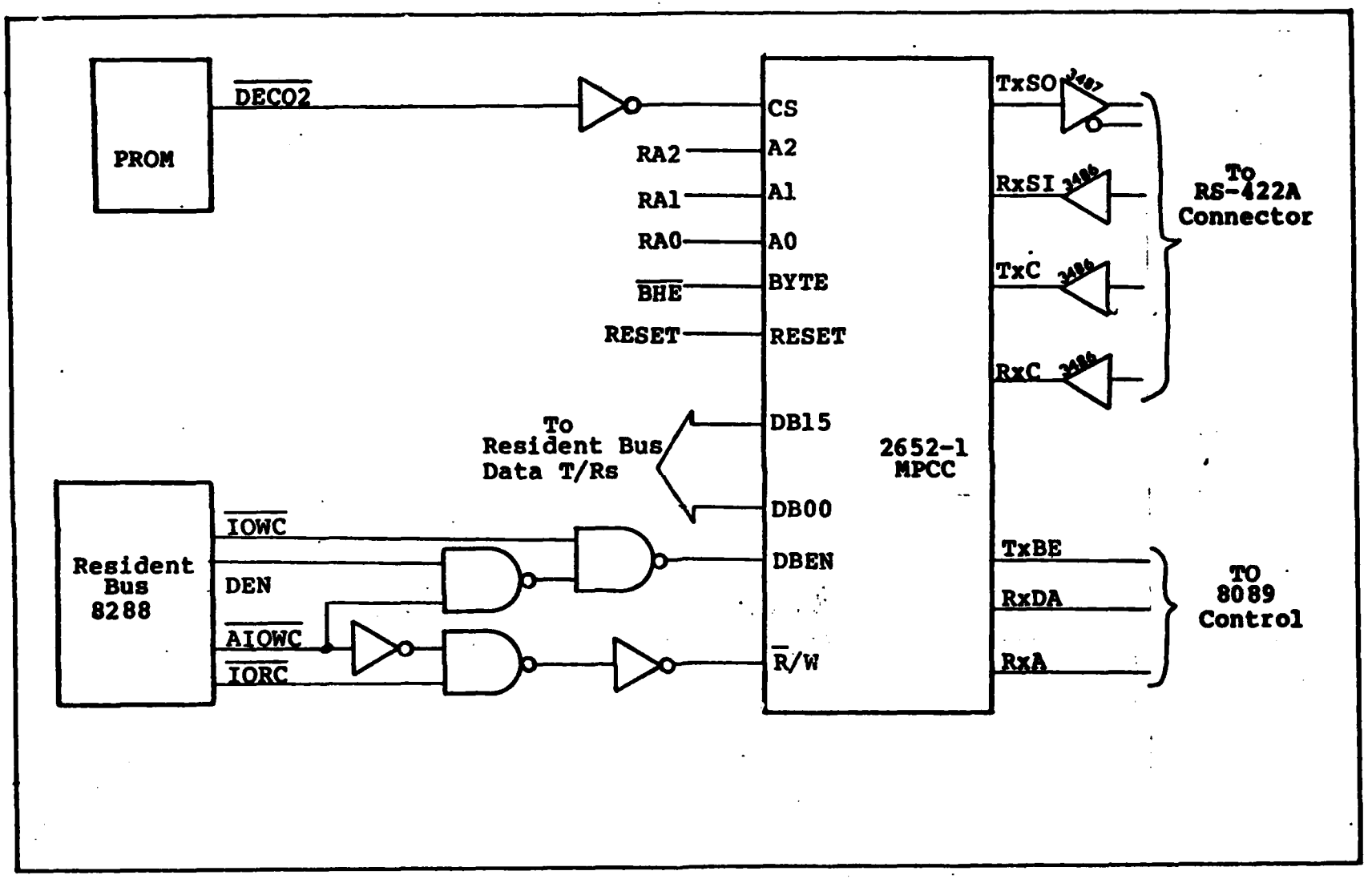


Figure 19. MPCC Control Signal Connections

bus cycle (Ref 20:B-17, B-79). The DEN positive transition occurs during the high clock period of T2 during a read cycle, which is a minimum of 60 ns after $\overline{\text{IORC}}$ is activated (assuming a 5 MHz clock). The $\overline{\text{IOWC}}$ signal is activated during the low clock period of T3, so it occurs a minimum of 155 ns after $\overline{\text{AIOWC}}$ is activated (Ref 20:B-79).

The MPCCs also have a maintenance mode (MM) selectable option. When the MM pin is grounded, the Transmitter Serial Output (TxSO) is gated back to the Receiver Serial In (RxSI), and the Transmitter Clock (TxC) is gated to the Receiver Clock (RxC) for off-line diagnostic purposes (Ref 18:820).

The receiver serial input and transmitter serial output are interfaced to the the computer network through RS-422A compatable line receivers and line drivers (Motorola MC3486 line receivers and MC3487 line drivers, respectively). The clock signals, RxC and TxC, can be provided over the RS-422A interface for synchronous communications or from a baud rate generator, such as an 8253 PIT, if asynchronous communications are being performed. The Receiver Data Available (RxDA) signal is routed to one of the 8089 channel's DRQ input for DMA source synchronization, and Receiver Status Available (RxSA) is connected to the IOP channel's respective EXT input to provide an external termination condition. The RxDA signal is activated when data is available in the MPCC's receiver buffer, and RxSA is activated when an end of message is received. For DMA

destination synchronization of the IOP during transmission, the Transmitter Buffer Empty (TxBE) signal is to be used as a condition for the synchronization of the IOP channel's DRQ input.

Construction of the Network Module

The network module was wire-wrapped on an Intel Prototype board. The physical layout of the board is shown in Appendix E. Dual-in-line package (DIP) sockets were used to allow all components to be plugged in, except for the 15-MHz crystal and the by-pass capacitors. The complete schematic for the demonstration network module is located in Appendix F. The .068uf by-pass capacitors, which are connected to each chip's Vcc pin, are not shown in the schematic diagram.

All TTL (transistor-transistor logic) components used are high-speed Schotky devices, except for the 7427 "Nor" gate and the 7485 comparators. Initially, two DIP sockets (B12 and C16) were left open to allow for circuit modifications. However, during testing of the network module (Chapter IV), it was necessary to use the socket at C16 for the 100 ohm resistors connected in-line with the CLK signal.

One edge of the prototype board has an Intel Multibus edge connector (86 pins) (Ref 20:A-198). The signal connections to this connector are shown in Figure 20. The interrupt lines (pins 35-41) were not being used, so the network module reset line was connected to $\overline{INT7}$ (pin 36). All signal or the Multibus are active low.

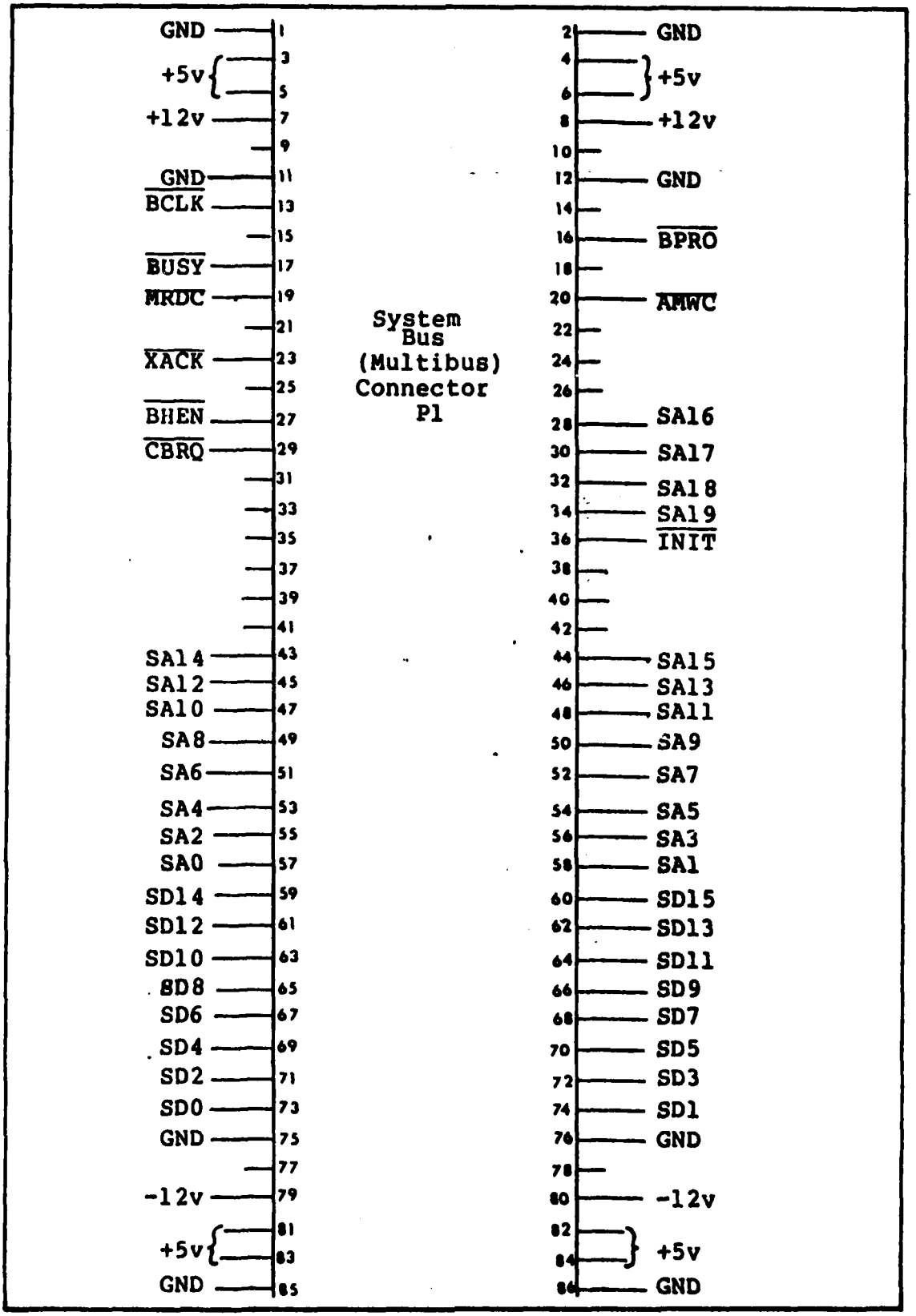


Figure 20. System Bus Connections

The two 2652-1 MPCCs used for the network I/O ports were not wired onto the network board. Space was left on the board for them to be installed. One of the MPCCs is to be installed where the 8251A USART is located. The $\overline{\text{DECO2}}$ signal can be inverted and used as the MPCC's chip enable. The chip enable signal for the other MPCC will need to be programmed into the PROM. The intended physical locations of the MPCCs are indicated by the dashed area on the physical layout diagram of the network board (Appendix E).

Summary of the Network Module Design and Construction

The design of the network module centers around the 8086 CPU and the 8089 I/O Processor. The 8086 and the 8089 are configured so they can access shared memory over a multi-master system bus and they can access both memory and I/O devices over a private (resident) bus. The 8089 operates as a slave to the 8086 and it performs all I/O operations including DMA transfers.. The 8089 executes the I/O channel program tasks specified by the 8086. Local arbitration (RQ/GT lines) allows them to share both the system bus and the resident bus.

An 8289 Bus Arbiter controls the access to the system bus. The RAM which is shared by the UNID II local and network modules is located on the local module and it is accessed by the network processors over the system bus. The network module has 8K bytes of resident RAM. The lower 1K bytes can be reserved for the 8086 interrupt vector table.

To demonstrate the ability of the network module to perform DMA transfers of data from shared memory to an output port, an 8251A USART was used for initial testing. In the final UNID II configuration, the USART will be replaced with two Signetics 2652-1 Multi-Protocol Communication Controllers. The 2652 was chosen for the following three major reasons (Ref 14:55): 1) the 2652 supports several bit-oriented and byte data link protocols; 2) both 8 and 16 bit word lengths can be accomodated; and 3) the 2652-1 supports transmission rates up to 2 Megabytes/Sec.

IV UNID II Test Procedure

This chapter presents the testing performed on the local module and the network module. Since an Intel SBC 86/12A single board computer (Ref 19) was being used as the local module, its testing was limited to verifying its proper operation after modifying its jumper selectable options (Ref 19:2-2). To satisfy the UNID II's functional requirements, it was necessary that the local module's jumpers be configured to provide an area of shared RAM memory which could be accessed by both the local and network modules. Also, the jumpers were configured to allow the network module to perform as the highest priority bus master (Ref 19:2-22).

The testing of the network module was more involved than that of the local module. Each section of the network module (i.e. decode logic, bus interfaces, enable logic) had to be tested seperately and its proper operation verified before proceeding to the next tests. (These tests were performed in a bottom-up fashion and are presented in further detail in this chapter). The network module was also tested to verify that it could properly access shared memory and perform DMA transfers of data from shared memory to an output port.

To aid in the testing of the UNID II, software test procedures were developed. These test procedures were written in PL/M 86 (Ref 28) and ASM 89 (Ref 20:3-44 to 3-

59). An ICE-86A in-circuit emulator (Ref 24) was also used for down-loading the test procedures from the Series II MDS (Ref 26) into the UNID II and for executing these procedures. The ICE-86A also provided the additional capabilities of executing single commands entered at the console, disassembling 8086 instructions, and performing single-step program execution while monitoring circuit-under-test operations.

This chapter is presented in the sequence in which the tests were performed. First, the hardware testing of the network module and the hardware testing of the local module will be presented. Then, the demonstration program which was developed to test the two modules is explained. Next, the software testing of the local module and the network module are presented. The UNID II test procedures and the results of these tests are then summarized.

Network Module Hardware Testing

Before construction of the network module began, both the system and the resident bus interfaces were wired on a bread-board. This was done so the bus control signals could be viewed on an oscilloscope, enabling a working knowledge to be acquired about the bus interfaces' operation before the actual "hard" wiring began. Initially the two 8288 bus controllers, their respective 8282/8283 address latches and 8286/8287 data transceivers, and associated output enable logic were breadboarded.

To provide the bus control signals, the bus controllers must decode a CPU's or IOP's $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$ status signals. The addition of a CPU or IOP to the breadboard would have complicated the bus interface testing, so a Hewlett Packard 8016A Word Generator (Ref 17) was used to generate timing signals which simulated the three status signals. An 8284 clock generator was also added to the breadboard to supply the 5-MHz CLK signals to the bus controllers. To observe the bus controllers' output signals, a Textronix 7704A oscilloscope with two 7A26 dual trace vertical amplifiers and a 7B53A dual time base was employed. The oscilloscope was connected to provide a time display of the bus controllers' output signals with respect to the status input signals.

The bus control signals observed on the oscilloscope are illustrated in Figure 21. The word generator was setup to provide 32-bit serial data at a bit rate of approximately 140-KHz. This set the pulse width of each bit to approximately 200ns. Relative to the status signals, all outputs appeared to be the same as specified in the manufacturer's timing diagrams (Ref 20:B-79).

The 8289 bus arbiter and the address decode logic for the SYSB/$\overline{\text{RESB}}$ signal were then added to the breadboard. The 7485 comparators were tested for several combinations of address inputs to ensure that the bus arbiter and the two bus controllers were being selected/deselected properly. Initially, the A<B and A>B outputs of the two comparators

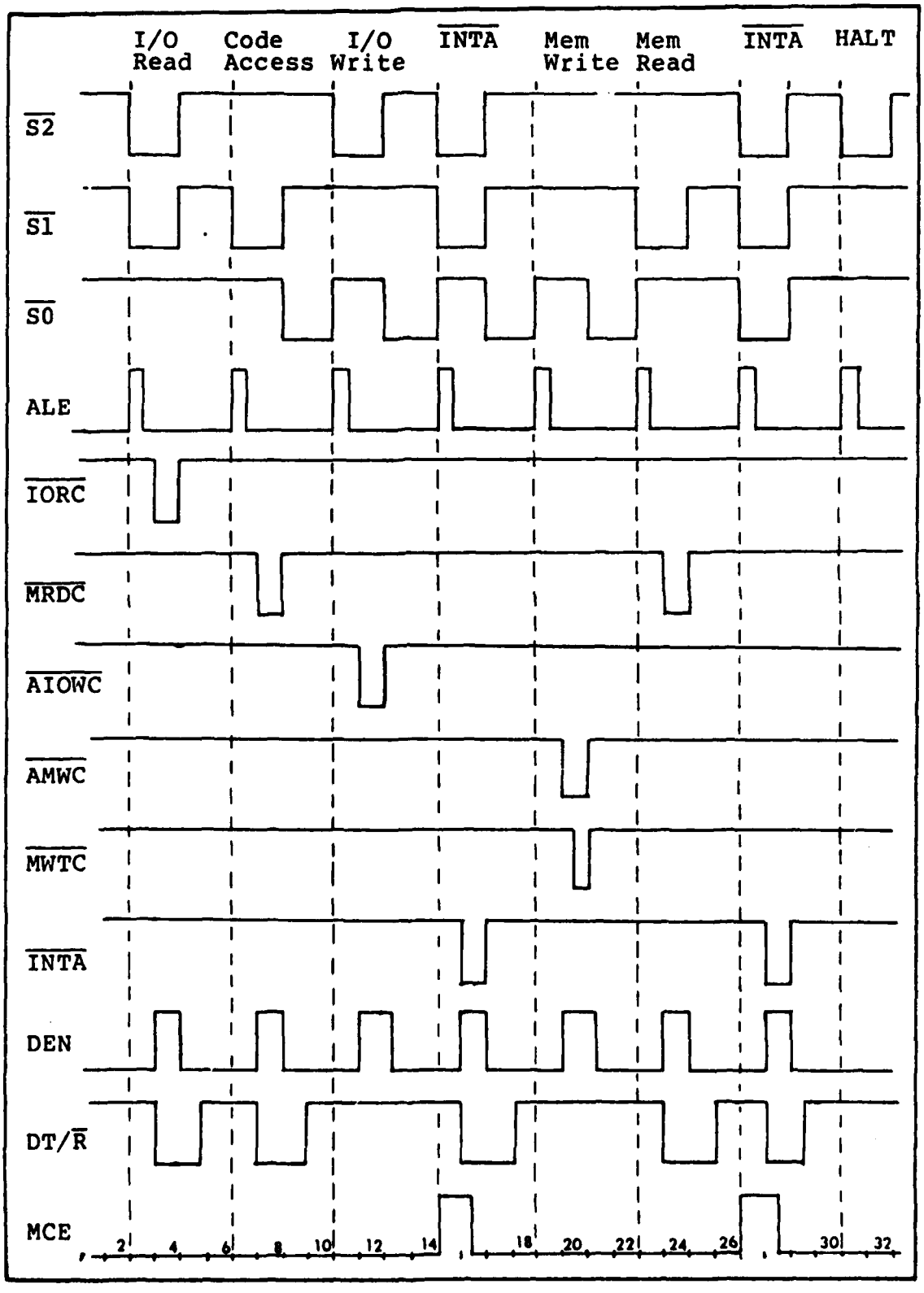


Figure 21. Bus Controller Output Signals

were interchanged, but this was corrected to the connections previously shown (Figure 16). To test the bus arbiter it was necessary to provide it with a bus clock ($\overline{\text{BCLK}}$) signal in addition to the CLK from the 8284. The $\overline{\text{BCLK}}$ pin was connected to the CLK line and the bus arbiter's output signals were observed to see if a bus access request was initiated. The bus request ($\overline{\text{BREQ}}$), common bus request ($\overline{\text{CBRQ}}$), bus priority out ($\overline{\text{BPRO}}$), and $\overline{\text{BUSY}}$ signals were all observed to go low when the SYSB/$\overline{\text{RESB}}$ input went high. This conforms to the bus arbiter's access request sequence (Ref 37:6-4). No timing relationship tests were made on the bus arbitration circuitry, since a 10-MHz $\overline{\text{BCLK}}$ was not being used.

Continuity Tests. After the network board was wire-wrapped, electricl continuity checks and visual inspections were made on the board. An audio continuity tester was used to check for wiring errors. The WLIST output (Appendix A) proved to be an invaluable aid during this and subsequent tests. A few wiring errors were detected and corrected.

Applied Power Testing. A 5vdc power supply was connected to the +5v and GND pins of the system bus connector. An ammeter was connected in series with the power connection to monitor the current being drawn by the network module. A voltmeter was also connected to the power terminals and both of these meters were monitored as each component was installed. The components were installed individually and tests were conducted as described in the following paragraphs:

Power was first applied to the network board with no chips installed. The ammeter verified that no current was being drawn by the module and a voltage check was made to each DIP socket pin which was to have been connected to +5v. The first component installed was the soldering terminal (E14) which contained the reset resistor and capacitor, and the reset switch. When this was installed the ammeter indicated that approximately 22 microamperes were being drawn. The current should have still been zero. The capacitor, C1 (Figure 12), was found to be effectively shorted (200K ohms). The capacitor was replaced and the proper zero current reading was obtained.

The next component installed was the 8284 clock generator (E17). The power supply current drain was within the manufacturer's specification. The CLK and PCLK outputs were traced with the oscilloscope to ensure they connected to the proper socket pins. The 8284 appeared to be functioning properly, but a high ripple component (approximately 400mv) was observed in the ground plane near the chip. A twisted pair of wires (CLK and GND) was connected between the clock generator and the resident bus controller, but this had very little effect on the ripple. A 22 guage wire was then connected directly from the system bus connector's GND (pin 86) to the 8284 GND pin. This lowered the ripple component down to approximately 50mv. Now, most of this ripple was probably due to insufficently shielded test leads and test setup, since some voltage variations were noted as the

oscilloscope test leads were touched.

The next components were installed in the following sequence:

- E10: 8288 Resident Bus Controller
- C10, C11, E11: 8282 Resident Address Latches
- A11, B11: Resident Data Transceivers

At this time the word generator and the oscilloscope were used to test the operation of the resident bus interface. This test was the same as conducted previously when breadboarding the bus interface components. The only error found during this test was that the data transceiver's (A21 and B21) output enable pins were connected directly to the DEN output of the bus controller. This would not work, because the DEN signal is active high and the output enables are active low. This was corrected by inverting the DEN signal before routing it to the data transceivers (Figure 15).

The order in which the next few chips were installed was as follows:

- B17: 74S288 PROM
- D14: Switch S1
- C14: C15: 7485 Comparator
- B15: 74S00 NAND
- A12: 8289 Bus Arbiter
- A11: 8288 Bus Controller
- A13: A14, A15: 8287 System Data Transceivers
- A16: A17, A18: 8283 System Address Latches
- B14: 7427 NOR
- B13: 74S04 Inverter
- B16: 74S32 OR
- B18: 74S32 OR
- D17: 7474 D-Type Flip-Flop

The word generator was used to supply the status signal timing to the system bus circuitry and to the bus arbiter. The word generator also supplied signals to simulate various other signals, such as $\overline{BHE}$, AD0, and address lines for testing the SYSB/$\overline{RESB}$ decode logic (Figure 16) and the system bus data transceivers' enable logic (Figure 15).

When testing the SYSB/$\overline{RESB}$ decode logic, it was discovered that it was necessary that the cascade inputs (pins 2, 3,and 4) of the 7485 comparators be grounded. Also, the truth table for the comparators had been misinterperted and the SYSB/$\overline{RESB}$ and CEN connections to the bus arbiter and bus controllers had to be rewired. The wiring was corrected to that shown previously (Figure 16).

Before testing the RAM and I/O device chip enables, the PROM (B17) had to be programmed. The details concerning the PROM programming are located in Appendix D. After the PROM programming was completed, the addresses of the chip select signals were verified.

Next, the MC 1488 line driver (D11) and the MC1489 line receiver (E11) were installed. The required +12 and -12 voltages were applied and the current drains were checked. At this time, the only chips remaining to be installed were the two 8K byte RAMs, 8086 CPU, 8089 IOP, 8253 PIT, 8259 PIC, and 8251A USART. These components were installed one at a time and their current and voltage levels were monitored. No excessive current was being drawn. This completed the hardware testing of the network module.

Local Module Hardware Testing

An Intel SBC 86/12A (Ref 19) is being used as the local module, so its testing was to involve only the executing of various commands in its ROM monitor. These tests were to include the checking of its RAM memory, the accessing of its I/O ports, and testing the operation of its system bus arbitration circuitry. However, when testing began, it was discovered that the EPROMs, which contained the monitor program, had been either inadverently erased or they had been replaced by empty EPROMS. A search was being made of backup disks to find the monitor listing when the 8086/88 In-Circuit Emulator (ICE-86A) (Ref 24) was received. By connecting the ICE-86A into the local module's 8086 socket, the board could be tested without the ROM monitor. The ICE-86A also performed downloading of programs from the ISIS-II Microcomputer Development System (Ref 26) to the local (or network) module.

Using the ICE-86A, the local module was tested by issuing I/O port commands to the on-board 8253 PIT and the 8255 Programmable Peripheral Interface (PPI) device. An ICE-86A demonstration program (Ref 24:3-1 to 3-43) was also compiled and downloaded into the local module. This helped in the testing of the local module and in learning the operation of the emulator. Notes concerning the use of the ICE-86A are located in Appendix G.

To use the SBC 86/12A as the local module, several jumper selectable options (Ref 19:2-2) had to be changed. The SBC

86/12A was initially wired in the factory default configuration, so all of the 32K bytes of its RAM memory were designated as being accessable by only the local processor. To permit the RAM to be shared with the network module, the E113-E114 jumper was installed and the switch, S1, settings were all opened. (All jumpers were installed with green wire-wrap; factory default jumpers are all blue). This configured the RAM memory to be mapped as shown in Figure 22. The RAM could now be shared with the network module. To the local module, the RAM is addressed between 00000H to 7FFFFH. To the network module, which must access this memory over the system bus, the RAM is mapped between locations 0F8000H and 0FFFFFH. Other jumper options are available to permit 8K byte blocks of the RAM to be dedicated for the local processor's use (Ref 19:2-8).

To configure the local module's 8289 bus arbiter to surrender the system bus to any other bus master (Ref 19:2-22), the E144-E145 and E130-E131 jumper connections were made. The E9-E26 and E134-E142 jumper connections were installed to permit the local module to output a reset pulse, over the system bus, to the network module (Ref 19:2-10,5-23). These jumpers connect the LSB of local module's PPI port "C" to the system bus $\overline{INT7}$ line.

Demonstration Program

To demonstrate and test the operation of the network module's bus circuitry, its DMA functions, and its accessing of shared and resident memory, a demonstration program was

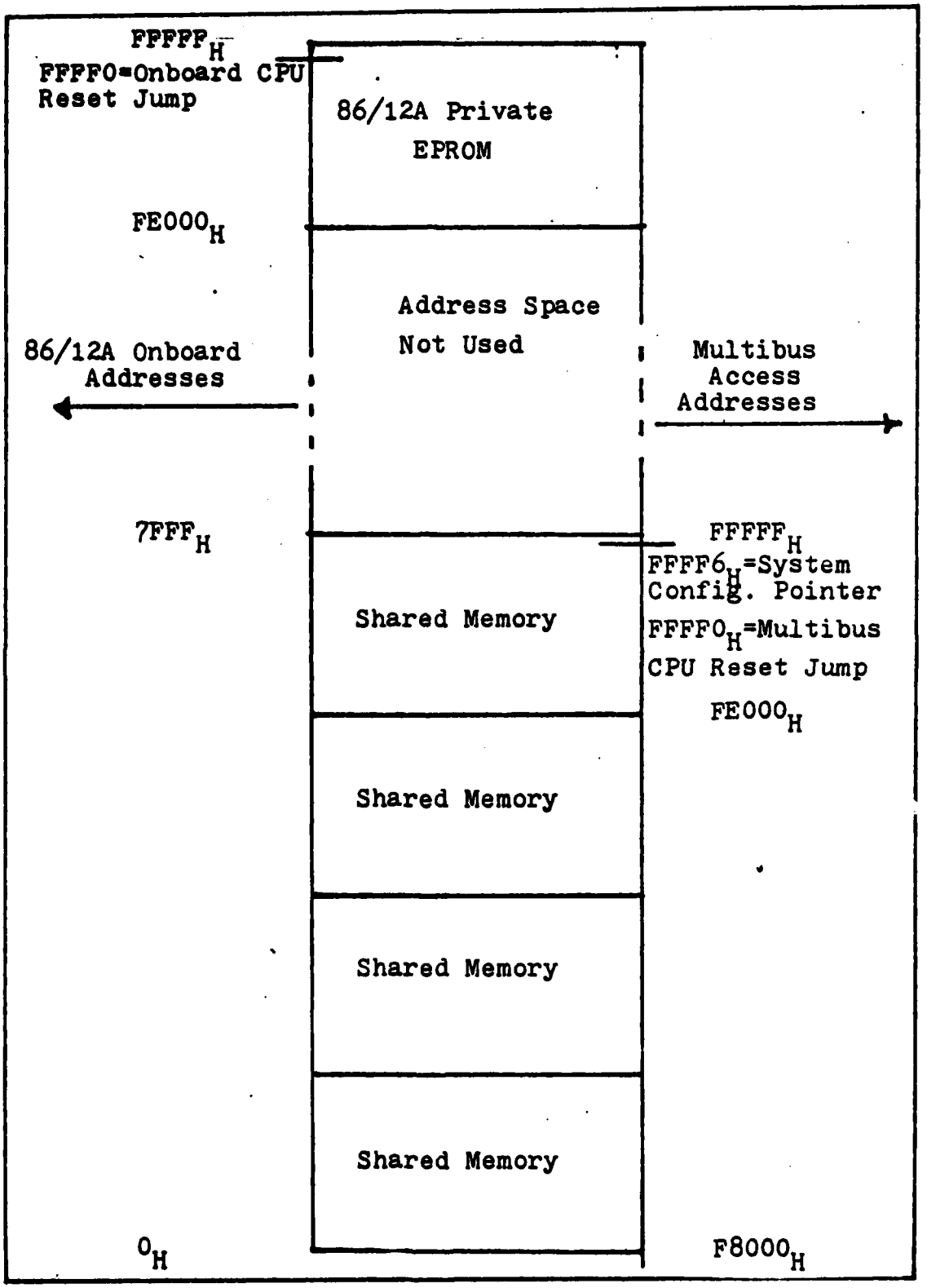


Figure 22. Memory Map of Shared Memory (Ref 30:3)

written in PL/M-86 (Ref 28) and in 8089 assembly language, ASM-89 (Ref 20:3-44 to 3-59). The demonstration program was developed to assist in verifying the proper design and construction of the network module and to provide a means of testing the functional operation of both modules. The demonstration program does not entirely verify that the UNID II meets all functional requirements. The tests were constructed to verify the operation of the shared memory and the network module and no local module I/O tests were performed.

An alternative to using the demonstration program for testing would have been to implement the UNID DELNET algorithmns being developed in a concurrent thesis project (Ref 15). However, this would have complicated the UNID II testing. These algorithmns would have to be converted from PL/Z (Ref 47) programming language to PL/M 86 and they could not be easily subdivided to provide individual testing of the various UNID II components.

The demonstration program is divided into three sections: first, an initilization procedure for the local module, called DEMOL; second, the network module's demonstration routines, DEMON; and third the 8089's TASK1 and TASK2 procedures. A source listing of this program is listed in Appendix H. This program was based on the prototype 8089 I/O system described in an Intel Application Note (Ref 30). However, the application note describes an 8089 system configured to operate in the remote mode and it does

not incorporate a resident bus, so several modifications of the basic program were necessary. Figure 23 shows the program flow of the demonstration program for the local module, DEMOL. The local module's functions are very simple for this demonstration and testing. First the local CPU intilizes its 8259A PIC, which in this case, will have all interrupts masked. Next, the CPU programs the PPI to send out a reset pulse to the network module (via INT7 of the system bus). Then the local processor enters a "forever" loop. This loop represents further processing by the local CPU since it can operate in parallel with the network processors.

The network module's program flow diagram is shown in Figure 24. After reset, the network CPU initilizes its 8259A PIC. All interrupts are masked except for IR00, which receives the interrupt from the 8089. Next, the CPU sets up the 8089 IOP's initilization blocks and issues the IOP its first Channel Attention (CA) pulse. While the IOP is performing its internal ROM initilization procedures, the CPU monitors the IOP's channel 1 busy flag. When the IOP completes its initilization, it clears the busy flag. The CPU detects this and continues its processing by setting up the 8089 communication blocks. This includes transfering the IOP's TASK1 program into the Task block.

The network CPU then issues the second CA to the IOP. When this CA is received, the IOP begins executing the TASK1 program in its task block. While the 8089 is executing its

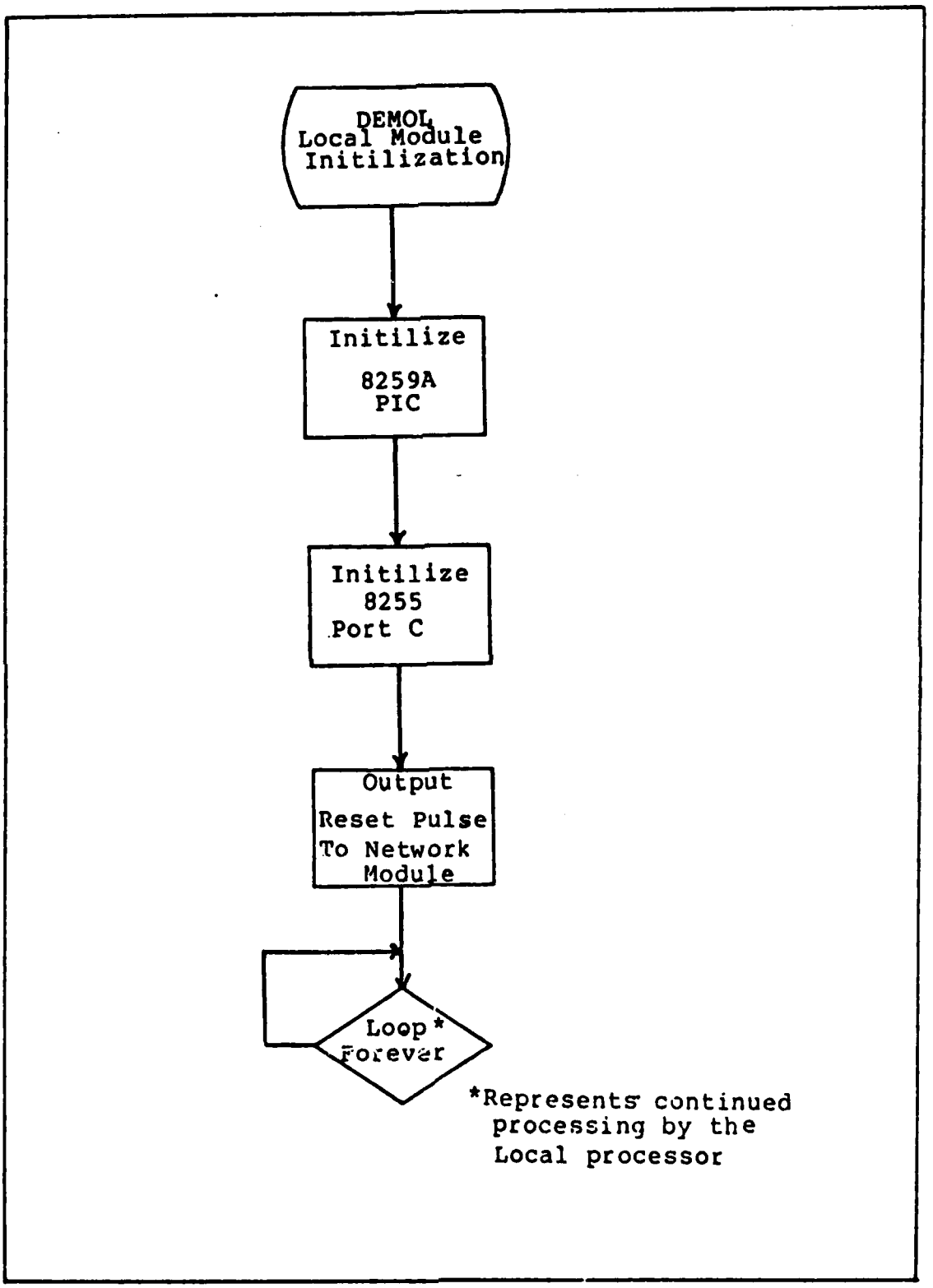


Figure 23. Flow Diagram of DEMOL

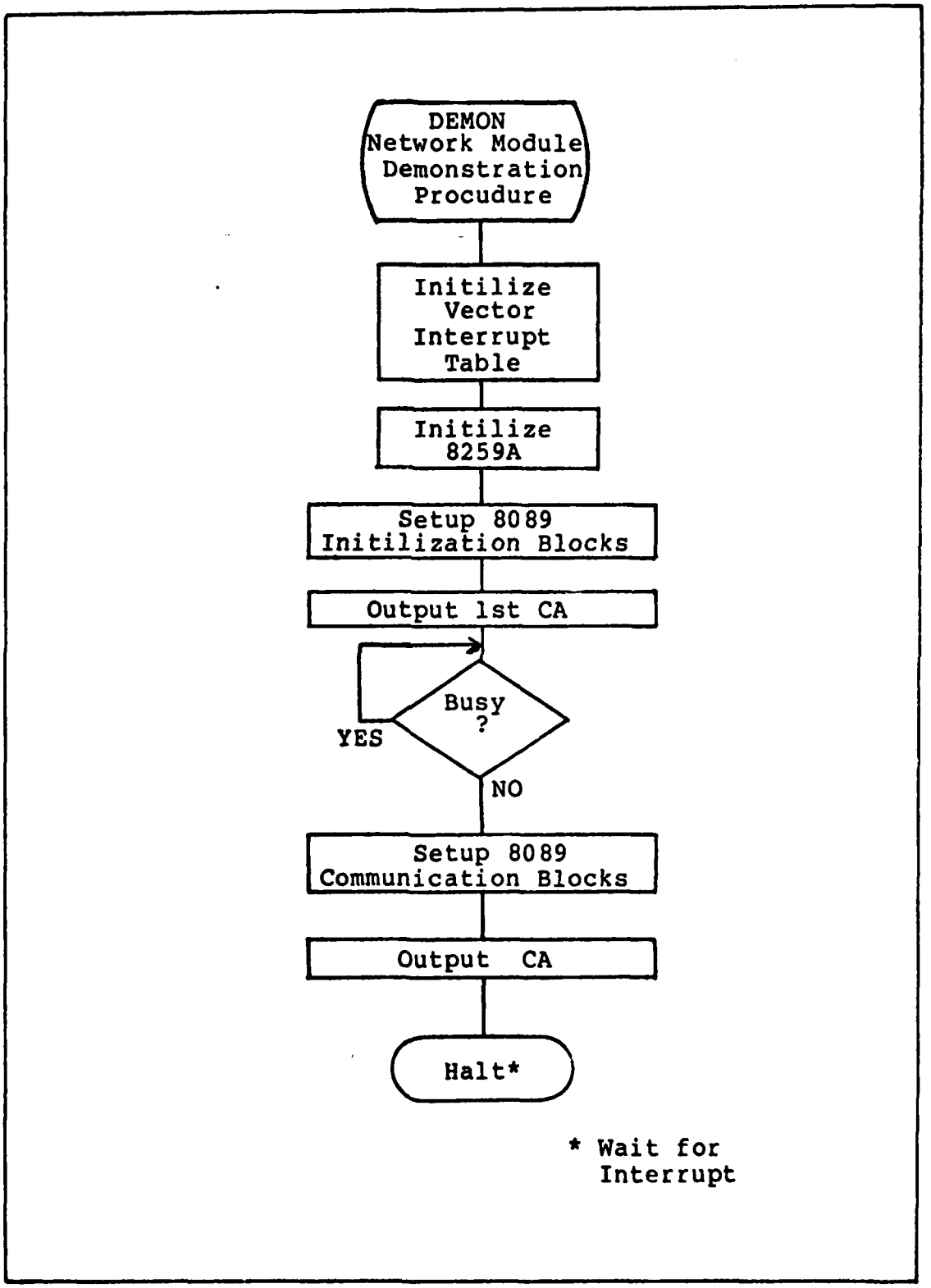
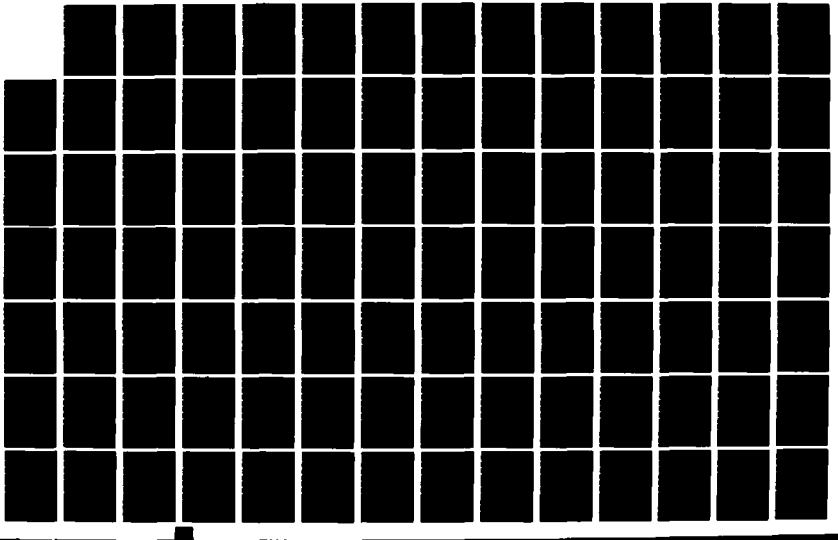
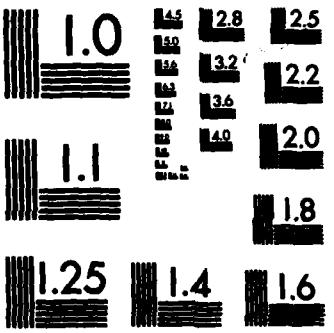


Figure 24. Flow Diagram of DEMON

AD-A124 834 DESIGN OF A PROTOTYPE UNIVERSAL NETWORK INTERFACE DEVICE USING INTEL 8086..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. D E PALMER 2/3

UNCLASSIFIED DEC 82 AFIT/GE/EE/82D-52 . F/G 9/2 NL





instructions, the CPU is halted with its interrupts enabled. (In normal operation a halt would not be necessary here; the CPU would enter into wait states if it had to perform a bus access while the 8089 had control of the bus). During Task1, the IOP initilizes the 8253 PIT and the 8251A USART. It also sets up the parameter block in which the intercommunication between the IOP and CPU takes place. (An 8089 Assembler was not available for assembling the TASK1 and TASK2 procedures, so they were entered into the DEMON program as data. They are listed in the source programs (Appendix H) as INITTB for TASK1 and PROGTB for TASK2).

When the 8089 has completed its TASK1, it issues an interrupt to the CPU. The IOP also surrenders the primary bus to the CPU via the $\overline{RQ}/\overline{GT}$ protocol. When the CPU receives the interrupt request, it acknowledges it and begins executing the message processing routine, MSGDSPL, shown in Figure 25. In this procedure, depending on the commands stored by the IOP in the parameter block, the CPU moves either a message or a menu to the message buffer. The CPU also moves TASK2 into the Task block. Then the CPU issues another CA to the IOP and halts.

When this CA is received, the IOP begins executing TASK2. The TASK2 program flow is shown in Figure 26. During TASK2, the IOP will DMA transfer the message buffer contents to the USART transmit data port. The USART will then transmit the data to the CRT. After the entire message buffer has been transferred, the IOP will poll the USART for

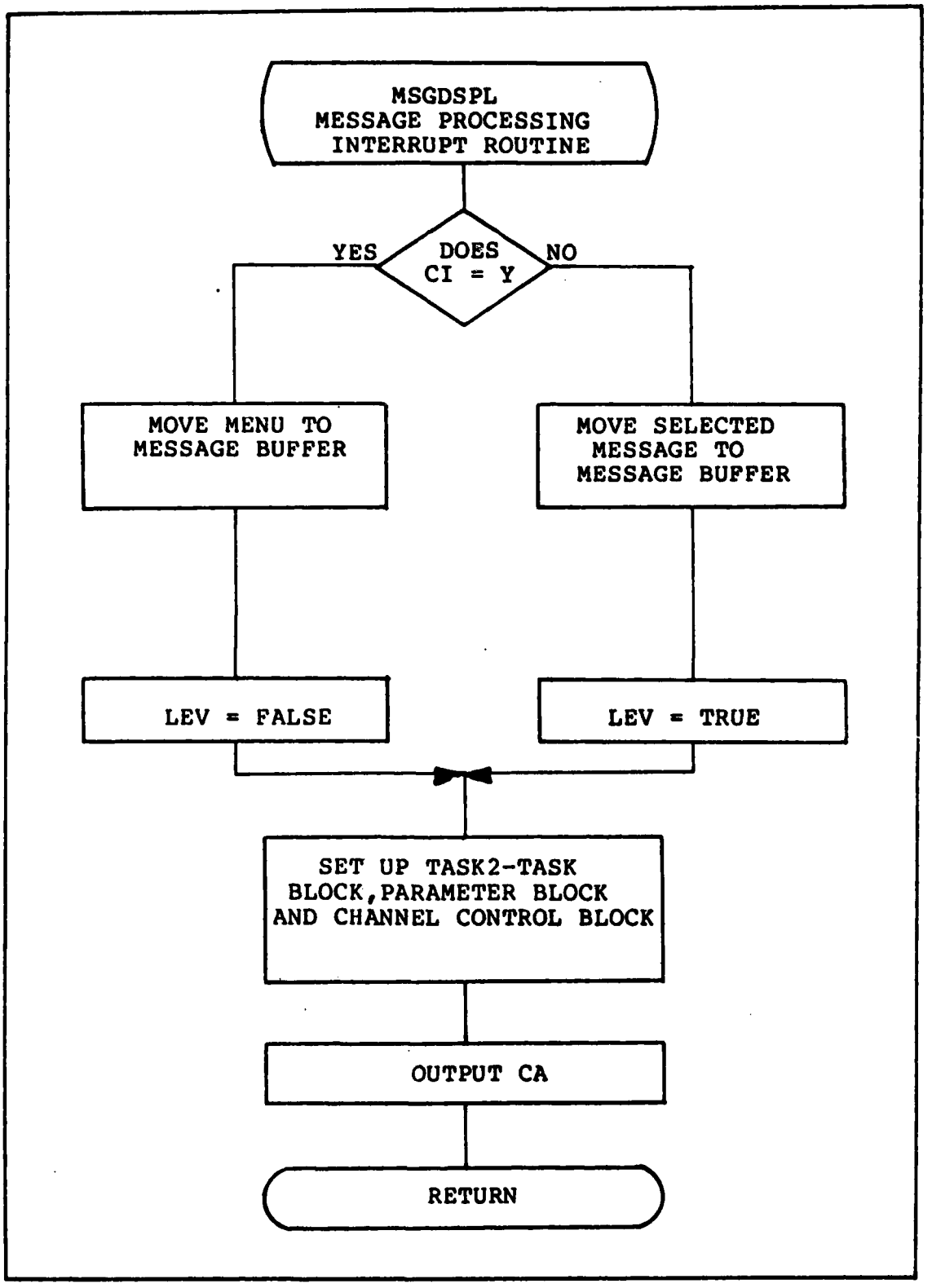


Figure 25. Flow Diagram of MSGDSPL (Ref 30:11)

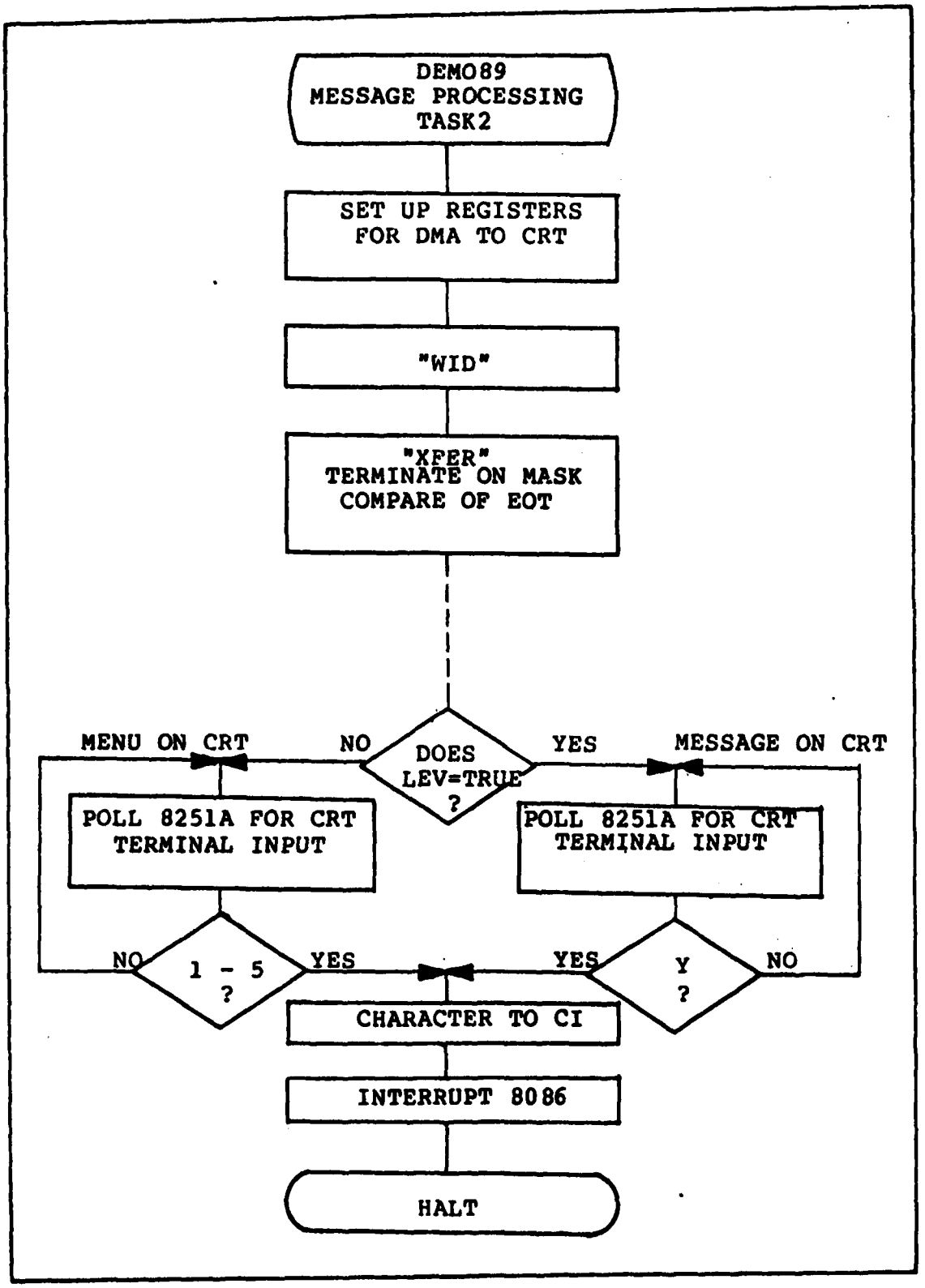


Figure 26. Flow Diagram of TASK2 (Ref 30:12)

valid input requests from the terminal. When a valid request (either for a message or the menu) is received, it is loaded into the parameter block and an interrupt request is issued.

When the interrupt request is received by the CPU, it again executes the MSGDSPL routine and then issues the CA to the IOP. The program execution sequence is now entirely interrupt-CA driven between the CPU and the IOP.

Local Module Software Testing

The ICE-86A was used to load the DEMOL program into the local module. A logic probe was connected to the system bus $\overline{\text{INT7}}$ line (pin 36) to detect the reset pulse from the local module to the network module. The reset pulse was output as expected. No other tests were performed on the local module.

Network Module Software Testing

The network module software testing began by installing the network module into the card cage without the local module. The ICE-86A was plugged into the network module's 8086 socket. The first tests performed were simple memory and I/O port access commands and a memory verify error kept occuring when a "write" was executed. A READY timing problem was suspected and a logic analyzer was connected to monitor several of the address/data lines and control lines. The logic analyzer was triggered on the memory read ($\overline{\text{MRDC}}$) and memory write ($\overline{\text{AMWC}}$) signals from the resident bus

controller. The error which was found was that the resident DT/$\overline{R}$ and DEN lines connected to the data transceivers were switched. This error was not detected previously since the mistake was made when entering the signals into the WLIST program.

After making the wiring corrections, the resident RAM could be accessed properly during both "read" and "write". However, a "control circuit failure" (Ref 24:B-1) began occuring at random times. The logic analyzer was again used to monitor the resident bus and READY control signals while memory access commands were issued. All signals appeared to be operating correctly. An oscilloscope was connected to the RESET line and the RESET was not going above 1.5 volts when the reset switch, S2, was pushed. There was a 1K ohm resistor in series with the switch and the 8284 $\overline{RES}$ input; this resistor was removed and the reset circuitry was wired as shown previously (Figure 12). This corrected the RESET circuitry and eliminated the control circuit failure error problem.

Another error was detected when the I/O devices were selected. They were being selected properly on their chip enable ($\overline{CE}$) lines, but they were hardwired to respond to both even and odd port addresses. Since they are 8-bit devices, they should only be connected to either an even or odd port, but not to both. The RA1 address line was connected to each of the I/O devices in place of the RA0 line to assign them even port addresses.

The next part of the network module testing involved the accessing of shared memory by the network processors. Both the network module (the 8089 was not installed) and the local module were installed into the card cage and connected to the system bus. A larger power supply had to be used, because together, the local module and the network module were drawning close to six amperes. The ICE-86A was plugged into the network module and an access of shared memory was attempted. Since the local module had not been initilized (DEMOL executed), the $\overline{INT7}$ line of the system bus was tied high. Also, the local module was strapped into a halt state ($\overline{S2}$=0,$\overline{S1}$=1,$\overline{S0}$=1).

The attempt to access shared memory was unsuccessful. With the aid of the logic analyzer, the system bus arbitration logic was checked and found to be operating properly. The shared memory decode circuitry on the SBC 86/12A was then checked and it was discovered that the PROM decoder (A69) was not being activated. The problem was traced to a bad address buffer latch (A13) on the network module. This chip was replaced and the access of shared memory was successful.

The DEMON program was complied (Ref 27), linked and located (Ref 29) to reside in shared memory begining at 0FC000H. This program was then downloaded into shared memory. The emulator was then used to single step through the 8086 instructions (the ICE-86A will not execute 8089 instructions). A "memory write failure" began appearing

again and the logic analyzer was used to trace the problem to another failure of the address latch, A13. The failure was caused by one of its output pins being tied to Vcc. A replacement was not available, so the SA16-SA19 address lines between the latch and the system bus were tied to ground (permanently enabled). This temporary modification is okay for initial testing, because the shared memory is mapped between 0F8000H-0FFFFFH and the SA16-SA19 address lines are always active when shared memory is accessed. The BHE/4B line, which was also routed through this chip, was rerouted through an unused latch on the resident bus and an inverter. When the replacement chip is received, the wiring connections can be returned to those shown in the WLIST.

During the emulation of the 8086 instructions there were two other chip failures which occured. One was that READY errors began occuring and they were traced to a bad 8284 clock generator. The other failure was a faulty NAND gate (B15) in the system bus data transceiver logic. It was discovered when the emulator began indicating an intermittent "memory write error" to system memory.

After all 8086 instruction had been successfully executed by the ICE-86A, the 8089 IOP was installed. To test if the 8089 was operating properly, the parameter block was checked after it had performed its initilization and other functions, and had given control back to the 8086 (ICE-86A). During the execution of TASK1, an oscilloscope was used to observe the output of the 8253 PIT, to ensure

that it was being initilized properly.

The emulation of DEMON worked fine through the 8089 initilization sequence and TASK1, but the CPU was not executing the MSGDSPL routine when the interrupt from the IOP was received. The logic analyzer was connected to the network module to monitor the interrupt acknowledge sequence. The problem found was that the resident bus data transceivers were being enabled at the same time that the 8259A PIC was to place the interrupt type onto the data bus. This was corrected by using the 8259A CE ($\overline{DECO3}$) and $\overline{INTA}$ as decode conditions for the data transceiver enables (Figure 15).

After these wiring corrections were made the DEMON program could be executed by the ICE-86A entirely. However, the message being displayed on the CRT had random characters missing. The DMA control logic was monitored with the logic analyzer, but every signal appeared to be timed properly. A 7474 was being used in the DMA control circuitry and this was replaced with a faster, 74S74 (Figure 18). This corrected the problem and the DEMON program appeared to be working properly. This completed the testing of the two modules.

Summary of UNID II Testing Procedure

This chapter presented the test procedure used in testing the local module and the network module of the UNID II. The hardware testing was performed on the network module in a bottom-up fashion. The proper operation af each

component (or each functional group of components) was verified before the testing was continued. Both power-off and power-on tests were performed on the network module. Since the local module was an off-the-shelf device, it did not require extensive hardware testing.

The demonstration program written to test the functional operation of the two modules was explained. The software testing of the UNID II involved the use of an ICE-86A to emulate the local and network 8086 processors. The test equipment used, the problems encountered, and their corrections were presented. A successful demonstration of the DMA transfer of data from shared memory to a network output port was achieved. This demonstration served as a test of the network module, including the system bus interface circuitry and the shared memory on the local module.

VI Conclusions and Recommendations

The objective of this investigation was to develop a Universal Network Interface Device (Unid II) based on the 16-bit architecture of the Intel 8086 family of microprocessors. The original UNID and the new version consist of two modules; a local module for interfacing to a host computer or a terminal, and a network module for interfacing to a computer network, such as the AFIT DELNET. The UNID II is intended to function as a node of a computer network. The 8086 UNID II system incorporates an off-the-shelf computer (SBC 86/12A) as the local local module. The network module uses an 8086 microprocessor and an 8089 I/O processor. Since no commercially manufactured boards exist with the required configuration, the network board had to been constructed.

A block diagram design had been completed in a previous thesis and based on this design, the network module, including the required control and interface circuitry was designed, constructed and tested. A demonstration network module was constructed first. It used an 8251A USART as an I/O port on the network module. The USART was used to aid in the testing of the network module. The interfaces to two Signetics 2652-1 Multiprotocol Communication Controllers were designed. These devices will be used to provide the required protocol and data conversions required at the UNID II/Network interface. Due to time limitations the MPCCs were not installed onto the

network board.

The network 8089 and 8089 were configured in a "local" mode of operation. In this mode, the 8089 performs all I/O operations including DMA transfers, mask/compare of data being transferred, and data translation of data, if required. All communications between the 8086 and 8089 is through a block of shared memory on the local module. The 8086 and the 8089 share the system bus and a resident bus, but they both cannot access it at the same time.

The software developed during this project was limited to that necessary for testing the internal functions of the UNID II and it external ports. No software was developed for controlling the local and network ports when connected as a node of a computer network.

Recommendations

The UNID II was constructed and tested with a minimum of components being used. To bring it up to full operational potential, another 8K of resident RAM could be installed. This RAM could be used to hold the communication blocks which are used by the 8086 and the 8089. This would eliminate most of the system bus accesses required, and the shared memory could then be dedicated to intermodule messages. Resident EPROMs could also be installed to store the network operating programs for the 8086 and the 8089. The MPCC devices will also need to be installed.

Algorithms for implementing the network higher level protocols are being developed in a concurrent thesis effort

(Ref 15). These algorithms are being written in PL/Z, and conversion of them to PL/M would be necessary for implementation in the UNID II. The ICE-86A would be an invaluable tool in this aspect of the development process.

The local module has only one on-board serial/parallel port and for connecting to a host and/or terminals, so there should be more installed. Commercial I/O boards are available (Intel SBC 508,SBC 517, and SBC 519 (Ref 21)), but they all connect to the Multibus, in which case the local processor would have to use the system bus for processing its messages in addition to the bus being used for inter-module messages. The local module has three parallel ports, with a total of 24 parallel input/output lines. A future project could investigate using these three ports for controlling three to five external serial-to-parallel I/O ports. One parallel port could possibly be used for data input, another for data output, and the third for the output of control signals. If the external ports are installed on a Multibus compatable prototype board, some of the auxillary connector pins could be used for additional control lines if required.

Additional support software will be required for further UNID II development; especially an assembler for the 8089 IOP. The 8089 test programs used in this project were entered in machine code, but they were very small routines. An iSBC 957 Intellec - iSBC 86/12A Interface and Execution Package (Ref 20:B-179) would bc helpful for down-

loading programs from the Intellec development system to the local module. The ICE-86A performs this function, but it is slow and it is necessary that the ICE-86A be plugged into either the local or network 8086 socket. Also, you cannot load a program, disconnect the ICE-86A and install an 8086 chip into the socket without risking damage to the devices.

The ROM monitor for the SBC 86/12A also needs to be replaced. I attempted to run the monitor by using the ICE 86A, but there appeared to be a bug in the monitor's source listing (listed under MON86.SRC). I recommend that the MON86.SRC program be recompiled using the DEBUG option (Ref 27:3-12) and then the ICE 86A be used to correct it. After all problems have been corrected, the monitor can then be programmed into EPROMs.

It is also recommended that a new Intel 16-bit microprocessor, the 80186 (also listed as an iAPX 86) (Ref 25), be investigated as a possible candidate for implementation on the network module. This 68 pin microprocessor is scheduled to be available commerically during the first quarter of 1983. The 80186 not only provides two times the throughput of the 5-MHz 8086, but it also has internal logic which replaces several of the support components used in 8086 systems. In addition to the bus interface unit and the execution unit, the 80186 on-chip logic includes the following (ICs typically replaced are in parenthesises):

- A 16-MHz clock generator. (8284 clock generator)

- Interrupt control logic which provides four external interrupts; three maskable and one non-maskable. Also, internal timer and DMA interrupts are provided. (8259A PIC not needed, but cascading possible).

- Three programmable counters; one is for internal event control only. (8253 PIT)

- Two seperate high-speed DMA channels which can transfer data at 2 Megabytes/Sec. (Could possibly replace 8089 IOP on the network module).

- Chip select logic. Six memory chip selects are provided for three address areas: upper memory, lower memory, and midrange memory. The 80186 can also generate chip selects for up to seven peripheral devices. (These could replace a PROM or several TTL components)

As can be seen from this list, the 80186 could possibly reduce the chip count of the network module while still maintaining present system functions. The ability to perform mask/compare and data translation during DMA transfers would be lost if the 8089 was replaced. However, the higher troughtput of the 80186 and reduced chip count of system components might be a greater advantage and warrents investigation.

Biblography

1. 1842 EEG/EEIC An Engineering Assessment Toward Economic, Feasible, and Responsive Base-Level Communications through the 1980's. Technical Report TR 78-5. Richard-Gebaur AFB, Missouri. October 1977.

2. Baker, Lee R. "Prototype and Software Development For Universal Network Interface Device." Unpublished Master's Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1980.

3. Bell, James R. "Future Directions in Computing," Computer Design 20(3): 95-102 (March 1981).

4. Bertine, H. V. "Physical Level Protocols," IEEE Transactions on Communications, COM-28 (4): 433-444 (April 1980).

5. Brown, Eric F. "Prototype Universal Network Interface Device." Unpublished Master's Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1979.

6. desJardins, Richard and George W. White. "ISO/ANSI Reference Model of Open Systems Interconnection," Proceedings of the Symposium on Computer Network Protocols. 47-53. Washington D.C. (May 29, 1980).

7. EIA Standard RS-232C. Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange. Washington, D.C.: Electronic Industrial Association, April 1975.

8. EIA Standard RS-449C. General Purpose 37-Position Interface for Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange. Washington, D.C.: Electronic Industrial Association, November 1977.

9. Electronic Design, Inc. Product Specification EDH8808. Ashland, Massachusetts: Electromic Design, Inc., May 1982.

10. Folts, Harold C. "A Powerful Standby Replaces the Old Interface Standby," Data Communications 9(5): 61-68 (May 1980).

11. ------. "Revised CCITT Recommendation X.25 -1980," Technical Information Bulletin 80-5, National Communications System, Office of Technology Standards Washington D.C., August 1980. (AD A092394)

12. Geist, John W. "Protocol Development for the Universal Network Interface Device." Unpublished Master's Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1981.

13. Grappel, Robert and Jack Hemenway. "Evaluating the 16-Bit Chips," Mini-Micro Systems XIII (12): 152-162 (December 1980).

14. Gravin, Andrew G. "Preliminary Design of a Computer Communications Network Interface Device Using Intel 8086 and 8089 16-Bit Microprocessors," Unpublished Master's Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1981.

15. Hazelton, Craig H. "Continued Development and Implementation of the Protocols for the Digital Engineering Laboratory Network," Unpublished Master's Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1982.

16. Hobart, William C. "Design of a Local Computer Network for the Air Force Institute of Technology Digital Engineering Laboratory, "Unpublished Master's Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, March 1981.

17. Hewelett-Packard. 8016A Word Generator, Operating and Manual Service Manual. Manual Order No. 08016-90003, Feburary 1977.

18. IC Master. (Vol. 1) Hearst Business Communications, Inc./ UTP Division. Garden City, New York, 1981.

19. Intel Corporation. "iSBC 86/12A Single Board Computer Hardware Reference Manual." Manual Order No. 9800645.

20. ------. The 8086 Family User's Manual. October 1979.

21. ------. "1977 Data Catalog," Santa Clara, California: Intel Corporation, 1977.

22. ------. "1981 Data Book," Santa Clara, California: Intel Corporation, 1981.

23. ------. iAPX 86,88 User's Manual. Manual Order No. 210201-001. July 1981.

24. Intel Corporation. "ICE-86A/ICE-88A Microsystems In-Circuit Emulator Operating Instructions for ISIS-II Users," Manual Order No. 162554-002, 1982.

25. ------. "iAPX186 - High Integration 16-Bit Microprocessor," Product Specifications, Order No. 210451-001 (May 1982).

26. ------. "ISIS-II User's Guide," Manual Order No. 9800306-05, 1979.

27. ------. "ISIS-II PL/M Complier Operator's Manual," Manual Order No. 980047.8A.

28. ------. "PL/M-86 Programming Manual," Manual Order No. 9800466A.

29. ------. "MCS-86 Software Development Utilities Operating Instructions for ISIS-II Users," Manual Order No. 9800639B.

30. Jigour, Robin. "Prototyping with the 8089 I/O Processor," Intel Application Note AP-89, May 1980.

31. McQullin, J.M. "Local Network Architectures," Computer Design 18 (5): 18-26 (May 1979).

32. Morse, Stephen P., William B. Pohlman and Bruce W. Ravenel, "The Intel 8086 Microprocessor: A 16-Bit Evolution of the 8080," Tutorial: Design of Microprocessor Systems (IEEE): 83-91 (1980).

33. Moulton, James. "Highlevel Protocol Boundaries in the ISO Model," Proceedings of the Symposium on Computer Network Protocols: 47-53. Washington D.C. (May 29, 1980).

34. Osborne, Adam. 8089 I/O Processor Handbook. Berkley, California: Osborne/McGraw-Hill, 1980.

35. Papp, Charles E. "Prototype DELNET Using the Universal Network Interface Device," Unpublished Master's Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1981.

36. Ravenscroft, Donald L. "Electrical Engineering Digital Design Laboratory Communications Network," Unpublished Master's Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1978.

37. Rector, Russell and George Alexy. The 8086 Book.

Berkley, California: Osborne/McGraw-Hill, 1980.

38. Schwartz, Mischa. Computer-Communication Network Design and Analysis. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1977.

39. Sluzevich, Capt Sam C. "Preliminary Design of a Universal Network Interface Device," Unpublished Master's Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1978.

40. Tanenbaum, Andrew S. "Network Protocols," Computing Surveys 13 (4): 453-489 (December 1981).

41. Tanenbaum, Andrew S. Computer Networks. Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1981.

42. Texas Instruments, Inc. The Semiconductor Memory Data Book for Design Engineers. Dallas, Texas: Texas Instruments, Inc., 1975.

43. Texas Instruments, Inc. The TTL Data Book for Design Engineers. Dallas, Texas: Texas Instruments, Inc., 1975.

44. Weinburg, Victor. Structured Analysis. New York: Yourdon Press, 1979.

45. Weissberger, Alan J. "Data-Link Control Chips: Bringing Order to Data Protocols," Electronics 51 (12): 104-112 (June 8, 1978).

46. Zilog, Inc. MCZ-1/20,25 Hardware User's Manual, Manufacturer's data. Cupertino, California: Zilog, Inc., 1977.

47. Zilog, Inc. PLZ User Guide (03-3096-01), Manufacturer's data. Cupertino, California: Zilog, Inc., July 1979.

48. Zimmermann, Hubert. "OSI Reference Model - The ISO Model of Architecture Open Systems Interconnection," IEEE Transactions on Communication, COM-28 (4): 425-432, (April 1980).

Appendix A

UNID II Data Flow Diagrams (Ref 11:26-34)

This appendix contains the Data Flow Diagrams (DFD) for the UNID II which were developed in a previous thesis effort. These DFDs show the UNID II message processing functions and the internal flow of messages between the local and network I/O ports. The Data Flow Diagrams are presented as follows:

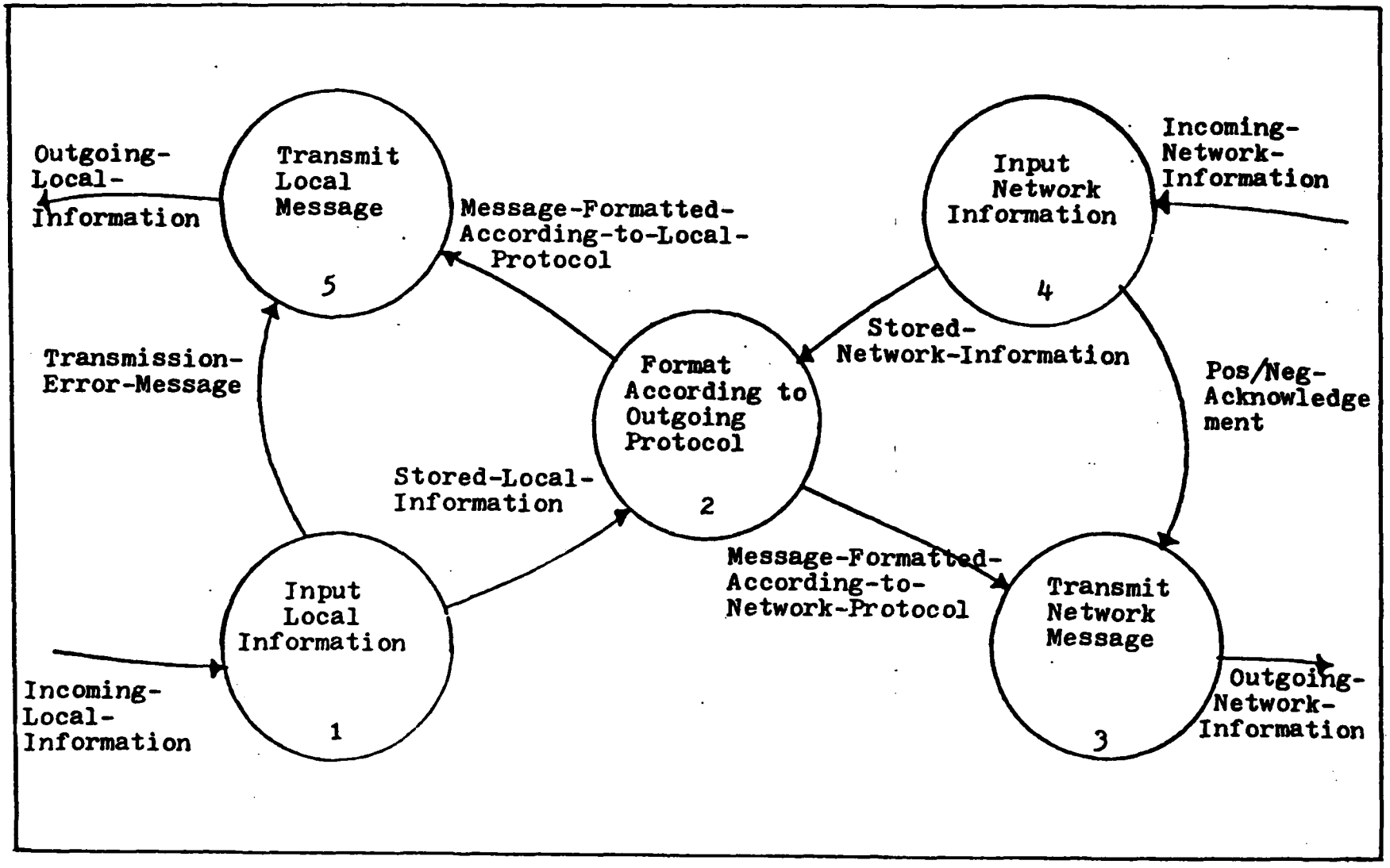


Figure A-1. UNID II Overview

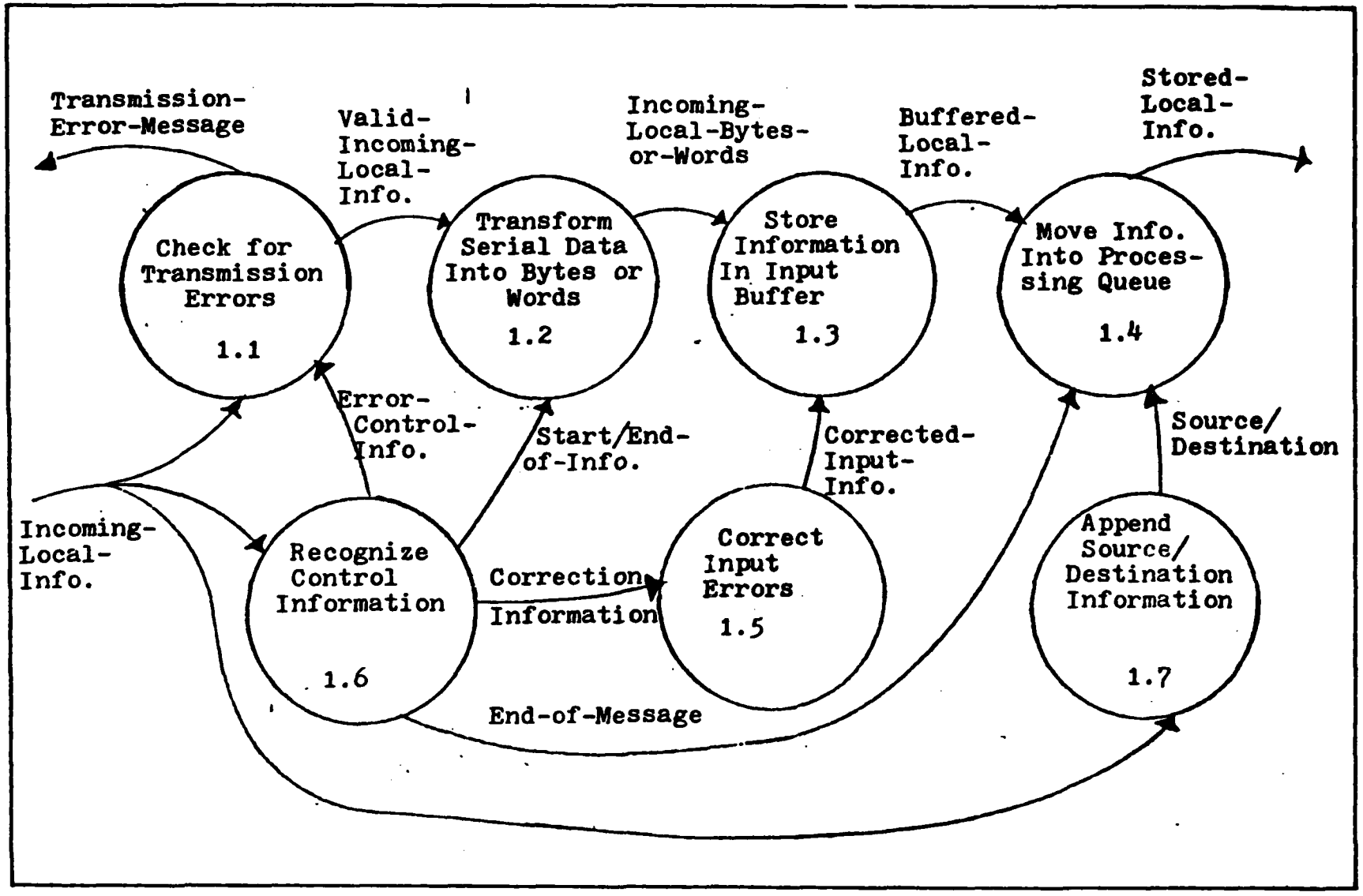


Figure A-2. Input Local Information

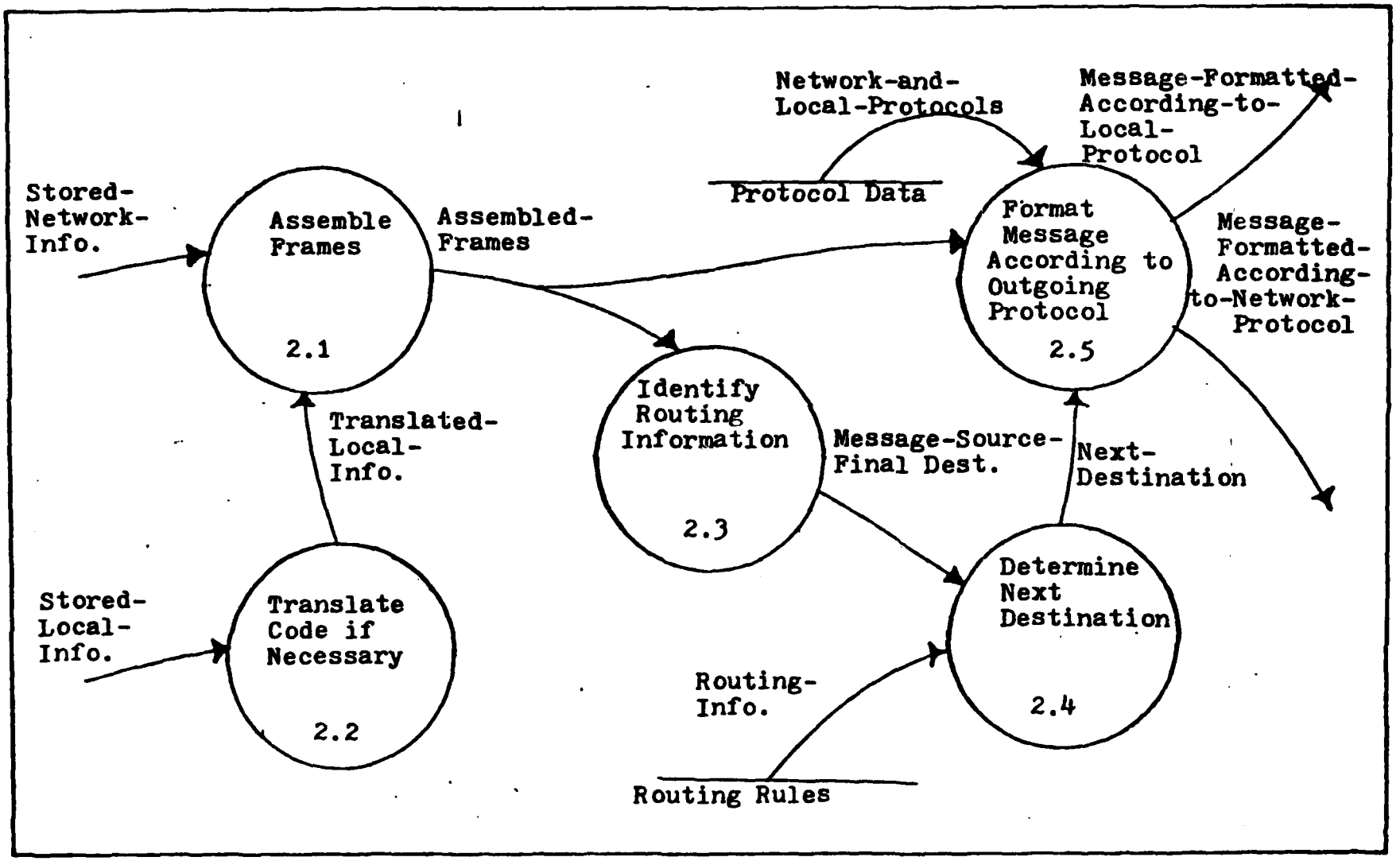


Figure A-3. Format According to Outgoing Protocol

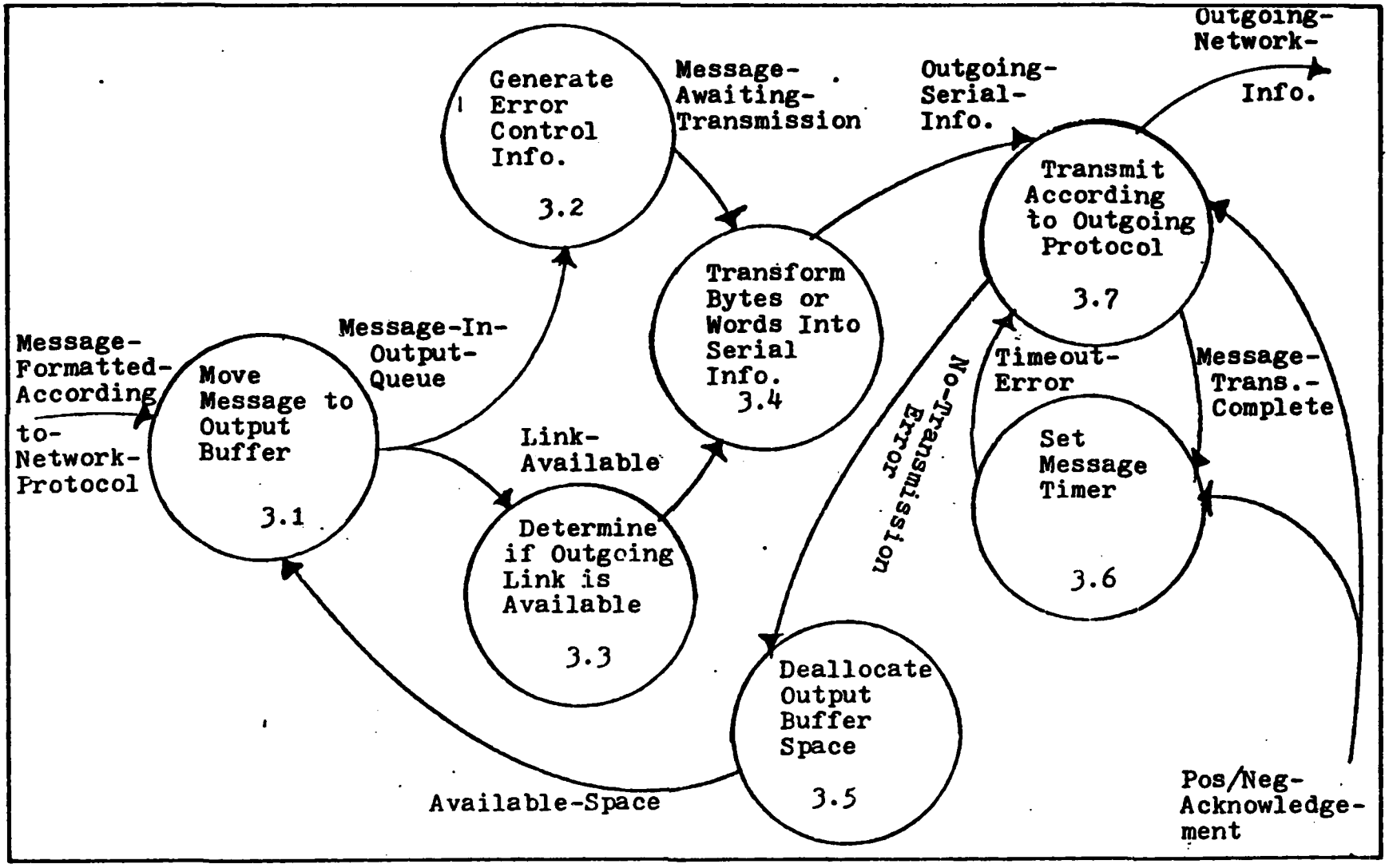


Figure A-4. Transmit Network Message

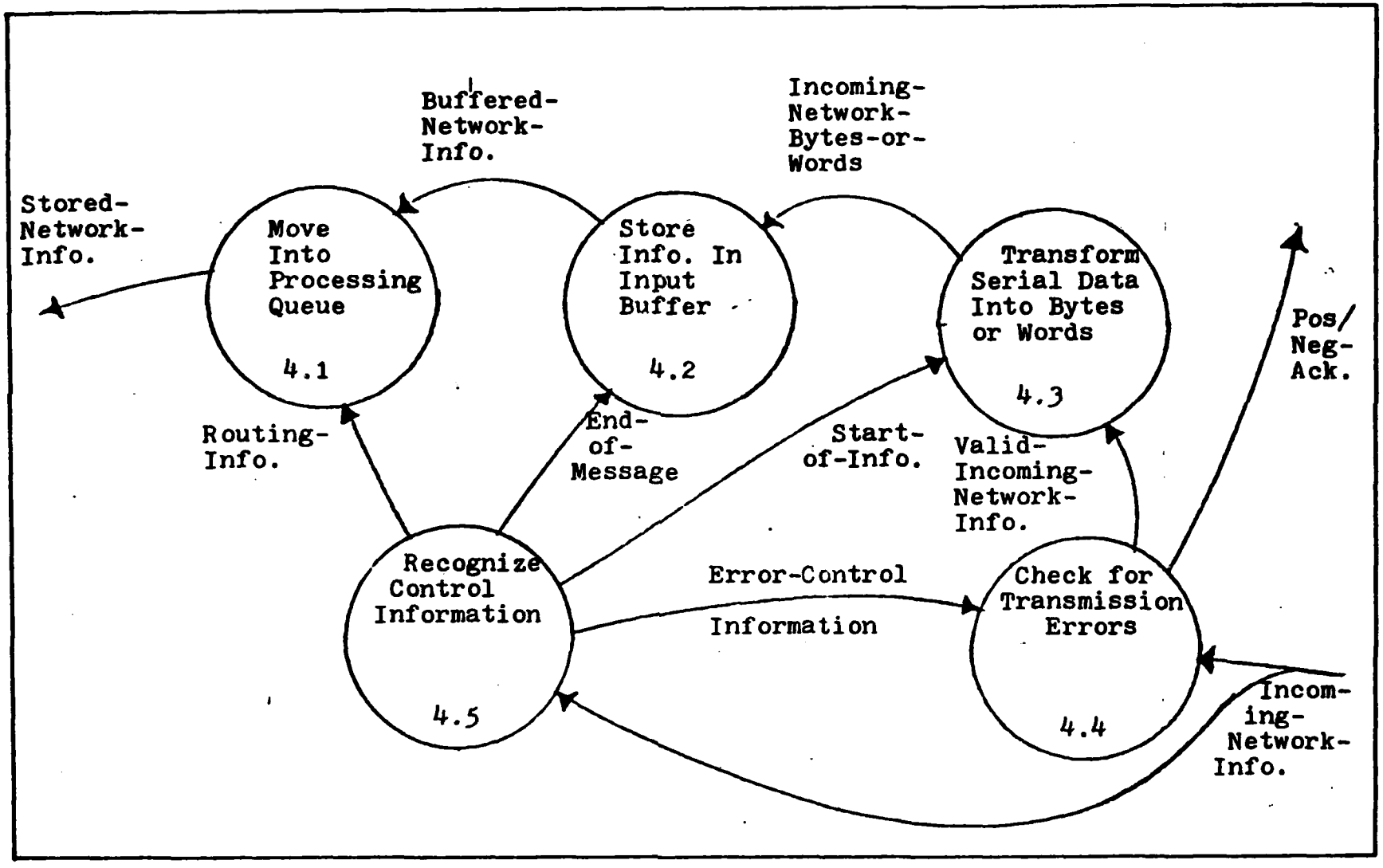


Figure A-5. Input Network Information

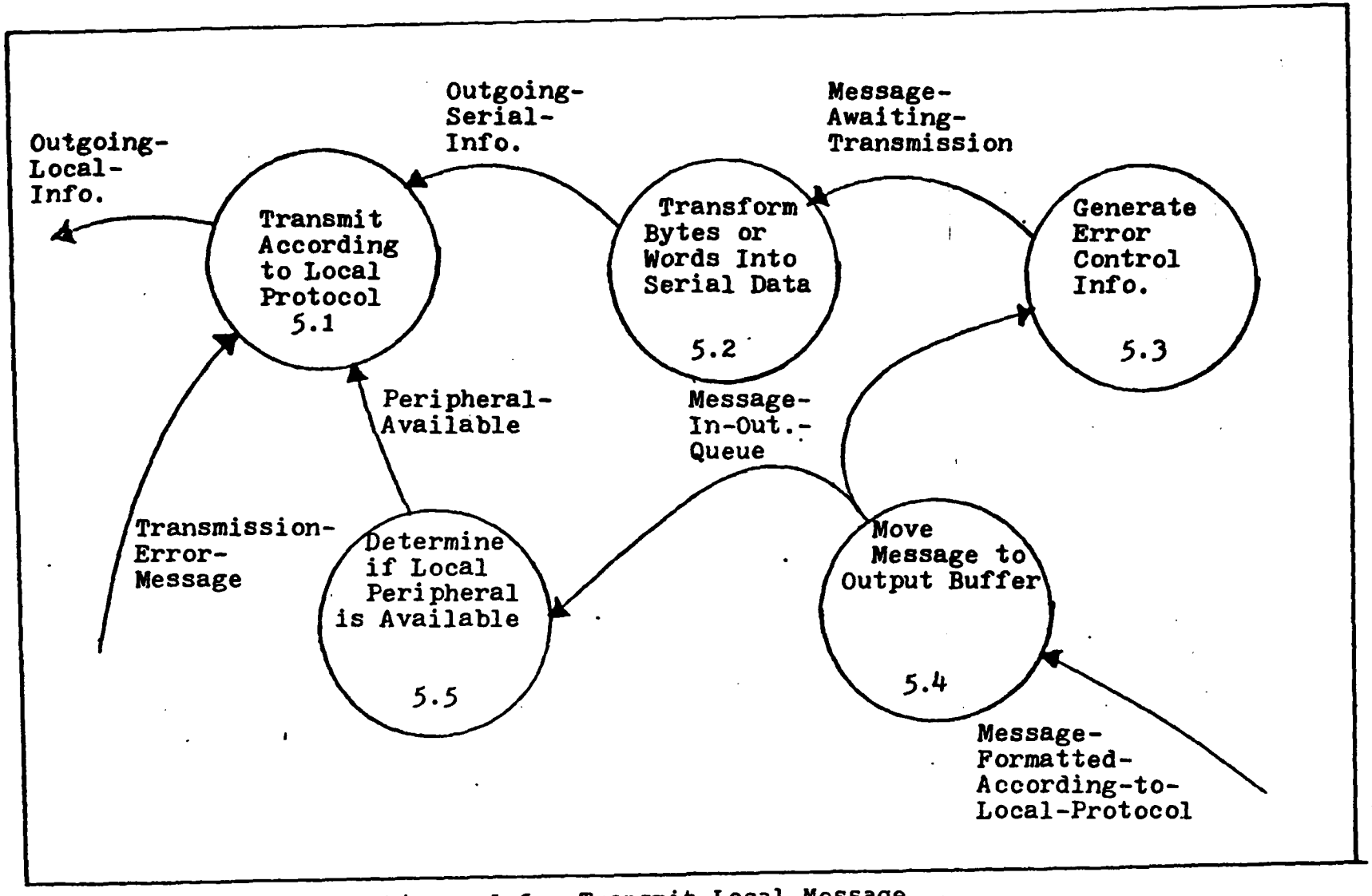


Figure A-6. Transmit Local Message

Appendix B

WLIST - Users Manual and Listing

This appendix contains the instructions for using the WLIST wire routing computer program. A card deck of the WLIST program has been obtained and they must be loaded into the computer (CYBER 750), if the program is not already installed. The program requires approximately 75000 units of memory (on the CYBER 750), so it cannot be run interactively. This appendix is seperated into the following sections:

Section B1

WLIST Users Manual

I. General Description

The purpose of the WLIST is to aid the hardware designer in generating an error free wire list, provide a part of the standard documentation of hardware design, and allow much simplified updating of documentation after a change in design has been made.

How it Works - The user provides a list of signal connections to each IC, plug, jack or other similar device. The data is sorted and collated in several different ways to provide the various output formats. An approximation of the best sequence of connections for each signal is generated. Errors of certain types can be detected and diagnostic messages are issued.

Output Forms - There are three output products. The first is essentially a reformatting of the input data. By unit (IC, etc.), the pin numbers and signal connected are listed. The second product is a list of all unit/pin connections by signal name, starting with the source of the signal. Also listed is the fan-out of each signal. The third product is a connection by connection wiring list, with seperate lists for each level of wire wrap.

II. How to Use the Program

Input Cards - There are three types of data cards; 1) the title card, 2) unit cards, and 3) connection cards. The title card contains an asterisk followed by a title of up to

20 characters. The title is printed on each page of the output listing.

example *PROM BOARD

Unit cards contain location, number of pins, and name (or comment) of each IC, connector, etc., on the board. One unit card is required per device. The first character is a dollar sign; followed by location (e.g. A:4), number of pins (1-999) and up to 10 comment characters.

example $A:1,24,EPROM 2708

Connection cards contain the names of signals to be connected to each pin of a device defined by a unit card. As many connections necessary follow each unit card. Signal names of up to 10 characters are listed, seperated by commas, starting with pin 1. Pins intended to have no connection must be named "NC". One signal source must be specified for each signal name used on a board. The source is identified by an asterisk preceeding the signal name. Connection cards may only use the first 72 columns. Columns 73 through 80 are reserved for optional sequence numbers.

example
```
$D:8,14,AND 7408
ENA1.CLK,*STB1,ENA2,CLK,*STB2,
GND,NC,NC,NC,NC,NC,NC,VCC
```

Executing the Program

WLIST can be run using a card reader or a time-sharing terminal. If run using a card reader, the following control card deck should be used.

```
RG1,CM73000,T35,IO35.E750511
ATTACH,WLIST,ID=E750511.
LIBRARY,COBOL.
COPY,INPUT,WLDATA.
WLIST.
7/8/9
data cards
6/7/8/9
```

If executed from a time-sharing terminal, the "deck" should contain the same control cards as above, with "*EOR" substituted for "7/8/9" and "*EOF" substituted for "6/7/8/9".

III. Interpreting the Output

Two header pages are printed identifying the circuit, version of WLIST, and the date and time at which the job was run. This allows immediate identification of the most recent run.

The normal output is self-explanatory. Signal sources are identified by a leading asterisk. The wire list output is provided in wire wrap levels. All level one connections should be made before preceeding to level two. This will eliminate wiring level changes and make later modifications of the circuit eaiser. Since the algorithmn for determining shortest string sequence is a simple one, the results may not always be the best possible. If there are signals that are sensitive to excessive wire length, their routing should be checked before the board is wired. Space is provided on the wire list to enter information such as wire guage and color or other appropriate remarks. The list is designed to be cut down to 8 X 10 1/2 size for easy use at the lab

bench.

Error Diagnostics - A moderate amount of error checking is done in the program. As with all such error diagnostics, care must be used in interpretation because the actual error may not be that indicated. The commonly encountered error messages will be discussed briefly.

If a signal name occurs only once, an informative diagnostic advises that the fan-out is zero. This may result from a typographical error in the string name.

If no source is declared for a signal, an informative diagnostic advises that there is no source. This may also indicate a typographical error.

If more than one source has been declared for a signal, an informative diagnostic advises such.

If any error has been detected, the wire list will be aborted and a fatal diagnostic will be printed. This is necessary because the results of the wire list are unpredictable if errors have been encountered.

IV. Conclusion

Although it will probably be more work for the designer to generate a wire list with this program, it has been found that it is useful in removing errors from the wire list and, in some cases, pointing out design errors. It forces more discipline on the designer, particularly in the often neglected area of documentation. It is of perticular advantage in keeping documentation up-to-date when changes are made in the design after fabrication. The few cards

affected can be changed, the program rerun, and a completely new, up-to-date set of documentation, without penciled corrections, is available. If done properly, use of the program can be well worth the time.

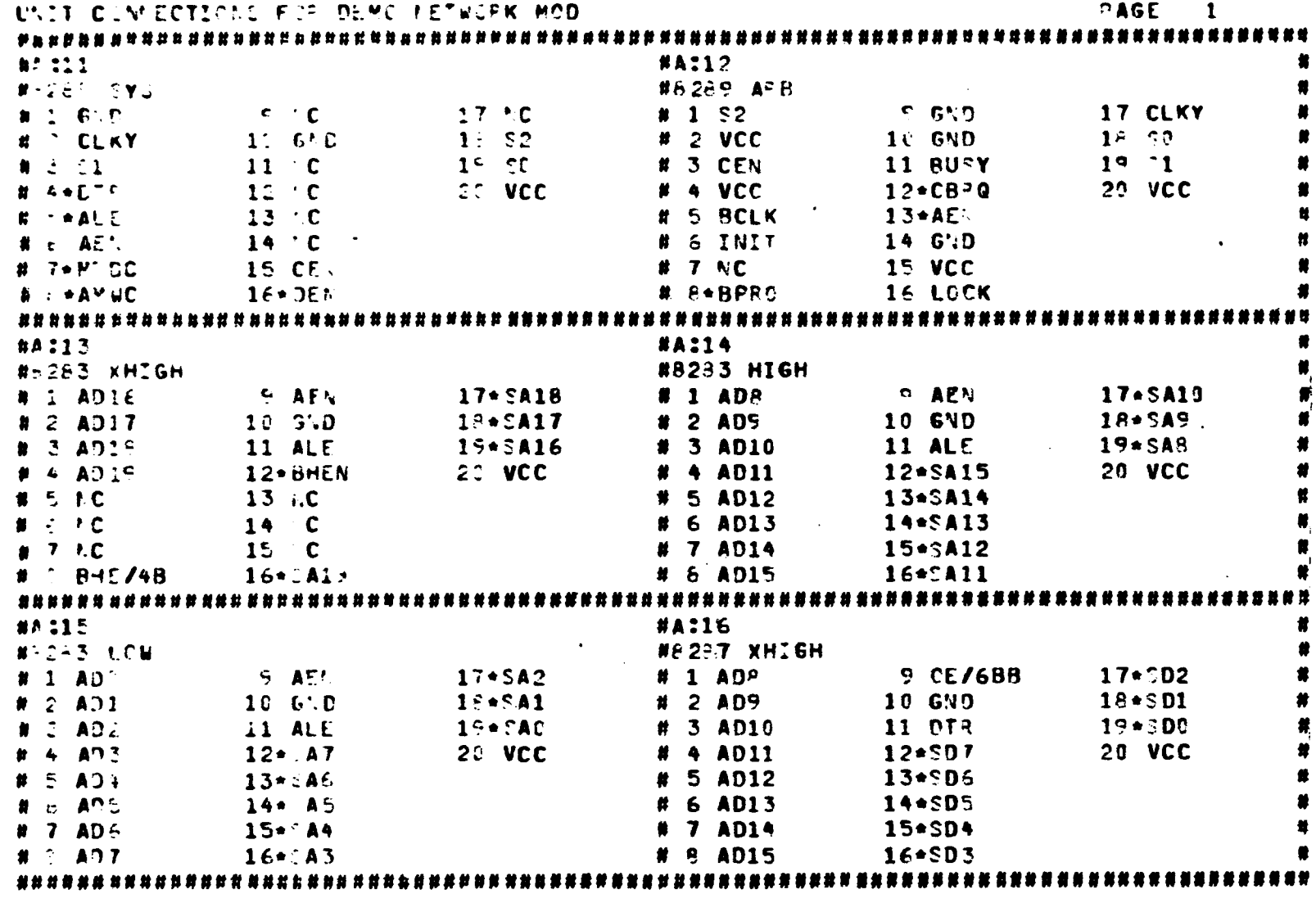
Section B2

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$WIRELIST PROGRAM              $$$$$
$$$$$VERSION 3.1-(MAY 18, 1978 )$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$WIRELIST FOR:                 $$$$$
$$$$$DEMO NETWORK MOD              $$$$$
$$$$$ 12/04/82  12.29.46.          $$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

I. Introduction

The wire listing for the demonstration network is included as part of this appendix on the following pages. The wire list is divided into three parts: 1) A reformat of the input data. Shown are each integrated circuit (IC) socket connections with pin numbers and signal names. The first number of each listing, e.g. A:11, refer to the IC socket's physical location on the wire-wrap board. 2) A list of all the signal names and their unit/pin connections. Also, the fan-out of each signal is given. 3) A connection by connection wiring list. There is a seperate listing for each level.

The WLIST program attempts to route all wires the shortest route between their connections and maintain a mimimum number of levels being required. This being the case, not all routes were followed on the level portion of the WLIST when wiring the network module. The IC sockets were numbered A11, A12, A13, etc. because the WLIST views A1 being physically closer to A11 and A12 than it is to A2. It should also be noted that the WLIST program does not route the Vcc and GND connections.



UNIT CONNECTIONS FOR DEMO NETWORK MOD PAGE 1

#A:11
#8288 SYS

1 GND	9 VCC	17 NC
2 CLKY	10 GND	18 S2
3 S1	11 VCC	19 S0
4*DEN	12 VCC	20 VCC
5*ALE	13 VCC	
6 AEN	14 VCC	
7*MEDC	15 CEN	
8*AMWC	16*DEN	

#A:12
#8289 APB

1 S2	9 GND	17 CLKY
2 VCC	10 GND	18 S0
3 CEN	11 BUSY	19 S1
4 VCC	12*CBRQ	20 VCC
5 BCLK	13*AEN	
6 INIT	14 GND	
7 NC	15 VCC	
8*BPRO	16 LOCK	

#A:13
#8283 XHIGH

1 AD16	9 AEN	17*SA18
2 AD17	10 GND	18*SA17
3 AD18	11 ALE	19*SA16
4 AD19	12*BHEN	20 VCC
5 NC	13 NC	
6 NC	14 NC	
7 NC	15 NC	
8 BHE/4B	16*SA19	

#A:14
#8283 HIGH

1 AD8	9 AEN	17*SA10
2 AD9	10 GND	18*SA9
3 AD10	11 ALE	19*SA8
4 AD11	12*SA15	20 VCC
5 AD12	13*SA14	
6 AD13	14*SA13	
7 AD14	15*SA12	
8 AD15	16*SA11	

#A:15
#8283 LOW

1 AD0	9 AEN	17*SA2
2 AD1	10 GND	18*SA1
3 AD2	11 ALE	19*SA0
4 AD3	12*SA7	20 VCC
5 AD4	13*SA6	
6 AD5	14*SA5	
7 AD6	15*SA4	
8 AD7	16*SA3	

#A:16
#8287 XHIGH

1 AD8	9 CE/6BB	17*SD2
2 AD9	10 GND	18*SD1
3 AD10	11 DTR	19*SD0
4 AD11	12*SD7	20 VCC
5 AD12	13*SD6	
6 AD13	14*SD5	
7 AD14	15*SD4	
8 AD15	16*SD3	

```
#A:17                                                    #A:18
#U2?7 HIGH                                               #U2?7 LOW
# 1 AD          9 .E/6BC      17*SD10      # 1 AD0        8 CE/6BC      17 SD2
# 2 AD-        10 GND         18*SD9       # 2 AD1       10 GND         18 SD1
# 3 AD10       11 DT?         19*SD8       # 3 AD2       11 DT?         19 SD0
# 4 AD11       12*.D15        20 VCC       # 4 AD3       12 SD7         20 VCC
# 5 AD12       13*SD14                     # 5 AD4       13 SD6
# 6 AD13       14* D13                     # 6 AD5       14 SD5
# 7 AD14       15*SD12                     # 7 AD6       15 SD4
# 8 AD15       16*SD11                     # 8 AD7       16 SD3

#A:19
#SC86/
# 1 GND         9*AD7        17 GND       25 NC         33 GND
# 2*AD14       10*AD6        18 INTR      26*S0         34*BHE
# 3*AD13       11*AD5        19 CLKX      27*S1         35*AD19
# 4*AD12       12*AD4        20 GND       28*S2         36*AD18
# 5*AD11       13*AD3        21 RESET     29*LOCK       37*AD17
# 6*AD10       14*AD2        22 READY     30 NC         38*AD16
# 7*AD9        15*AD1        23 NC        31 RQ/GT0     39*AD15
# 8*AD8        16*AD0        24 NC        32 NC         40 VCC

#A:21
#U2?6 HIGH
# 1 AD-         9 DEN        17*RD10
# 2 AD9        10 GND        18*RD9
# 3 AD10       11 DTR/R      19*RD8
# 4 AD11       12*.D15       20 VCC
# 5 AD12       13*RD14
# 6 AD13       14*RD13
# 7 AD14       15*RD12
# 8 AD15       16*RD11
```

#A:22
AMHC 8KX
1 NC 9 MA2 17 MD13 25 MA9
2 MA13 10 MA1 18 MD14 26 NC
3 MA 11 MD8 19 MD15 27 AMWC/R
4 MA7 12 MD9 20 HIMAM 28 VCC
5 MA6 13 MD10 21 MA11
6 MA5 14 GND 22 MRDC/R
7 MA4 15 MD11 23 MA12
8 MA3 16 MD12 24 MA10

#B:11
#8253 TIMER
1 AD7 9 CLK2 17 NC
2 AD6 10*OUT2 18 NC
3 AD5 11 GND 19 MA1
4 AD4 12 GND 20 MA2
5 AD3 13 NC 21 DEC01
6 AD2 14 NC 22 IORC/R
7 AD1 15 NC 23 AIOWC/R
8 AD0 16 NC 24 VCC

#B:12
#NOT USED
1 NC 9 NC
2 NC 10 NC
3 NC 11 NC
4 NC 12 NC
5 NC 13 NC
6 NC 14 NC
7 NC
8 NC

#F:13
#74S04 INV
1 NC 9 NC
2 NC 10*CA
3 NC 11 CAX
4 NC 12*TXINV
5 XACK 13 TXRDY
6*RDY1 14 VCC
7 GND
8 NC

#B:14
#7427 NOR
1 DEC02 9 TXINV
2 AIOWC/R 10 FOOUT
3 BHE 11 GND
4 AD0 12*FFCLK
5 GND 13 MA1
6*BHE/4B 14 VCC
7 GND
8*DRQ1

#B:15
#74S00 NAND
1 L1 9 MA0
2 L1 10 BHEN
3*L2 11*DENINV
4 ALTB 12 DEN
5 AGTB 13 DEN
6*CEN/R 14 VCC
7 GND
8*L1

```
#B:16                          #B:17                          #B:18                          #
#74S32 OR                      #PROMDECODE                    #74S32 OR                      #
# 1 RDYEC        8 DEMINV      # 1*DEC00        9 NC          # 1 RHEL         9 DEC04       #
# 2 RDYM        10 L2          # 2*DEC02       10 RA12        # 2 DEC00       10 AIOWC/R     #
# 3*RDYX        11*RDY2        # 3*DEC03       11 RA13        # 3*HIRAM       11*OEF         #
# 4 L1          12 INTAINV     # 4*DEC01       12 RA14        # 4 DEC00       12 OEDI        #
# 5 DEMINV      13 RDYX        # 5*DEC04       13 RA15        # 5 RA0         13 DDEN/R      #
# 6*CE/6BB      14 VCC         # 6 NC          14 CEN         # 6*LOWRAM      14 VCC         #
# 7 GND                        # 7 NC          15 GND         # 7 GND                        #
# 8*CE/6BC                     # 8 GND         16 VCC         # 8*CAX                        #
#B:21                                                                                        #
#2716 LOW                                                                                    #
# 1 AD0          9 OEF         17*RD2                                                       #
# 2 AD1         10 GND         18*RD1                                                       #
# 3 AD2         11 DTR/        19*RD0                                                       #
# 4 AD3         12* D7         20 VCC                                                       #
# 5 AD4         13*RD6                                                                      #
# 6 AD5         14* D5                                                                      #
# 7 AD6         15* D4                                                                      #
# 8 AD7         16* D3                                                                      #
#B:22                                                         #C:14                          #
#AMLS 8KX                                                     #7485 COMPA                    #
# 1 NC           9 RA2         17 RD5        25 RA9         # 1 SS1S4        9 SS1S1       #
# 2 RA13        10 RA1         18 RD6        26 NC          # 2 GND         10 RA16        #
# 3 RA8         11 RD0         19 RD7        27 AMWC/R      # 3 GND         11 SS1S2       #
# 4 RA7         12 RD1         20 LOWRAM     28 VCC         # 4 GND         12 RA17        #
# 5 RA6         13 RD2         21 RA11                      # 5*AGTB        13 RA18        #
# 6 RA5         14 GND         22 MPDC/R                    # 6 NC          14 SS1S3       #
# 7 RA4         15 RD3         23 RA12                      # 7 NC          15 RA19        #
# 8 RA3         16 RD4         24 RA10                      # 8 GND         16 VCC         #
```

```
UNIT CONNECTIONS FOR DEMO NETWORK MOD                                          PAGE   5
##########################################################################################
#C:15                         #C:16                         #C:18                        #
#74F5 COMPB                   #LOADRESTRS                   #74S00 NAND                  #
# 1 SS198       9 SS195       # 1 NC          9 CLK         # 1 AIOWC/R      9 DEC03     #
# 2 GND        10 RA16        # 2 NC         10 CLK         # 2 IORC/R      10 DEC01     #
# 3 GND        11 SS196       # 3 NC         11 NC          # 3*RDYIC       11*CEN       #
# 4 GND        12 RA17        # 4 NC         12 NC          # 4 AMWC/R      12 CEN/R     #
# 5 NC         13 RA18        # 5 NC         13 NC          # 5 MRDC/R      13 CEN/R     #
# 6 NC         14 SS197       # 6 NC         14 NC          # 6*RDYM        14 VCC       #
# 7*ALEB       15 RA19        # 7*CLKY       15 NC          # 7 GND                      #
# 8 GND        16 VCC         # 8*CLKX       16 NC          # 8*OEDIS                    #
##########################################################################################
#C:20                                       #C:21                                        #
#8282XHIGH                                  #8282 HIGH                                   #
# 1 BHE         9 GND        17 NC          # 1 AD8          9 GND        17*RA10        #
# 2 NC         10 GND        18 NC          # 2 AD9         10 GND        18*RA9         #
# 3 NC         11 ALE/R      19*BHEL        # 3 AD10        11 ALE/R      19*RA8         #
# 4 NC         12*RA19       20 VCC         # 4 AD11        12*RA15       20 VCC         #
# 5 AD16       13*RA18                      # 5 AD12        13*RA14                      #
# 6 AD17       14*RA17                      # 6 AD13        14*RA13                      #
# 7 AD18       15* A16                      # 7 AD14        15*RA12                      #
# 8 AD19       16 NC                        # 8 AD15        16*RA11                      #
##########################################################################################
#C:24                         #D:11                         #D:14                        #
#74S00 NAND                   #MC1489 REC                   #SWITCH                      #
# 1 NC          9 INTA/R      # 1 REC232      9 NC          # 1 GND          9*SS1S1     #
# 2 NC         10 INTA/R      # 2 NC         10 NC          # 2 GND         10*SS1S2     #
# 3 NC         11*DDEN/R      # 3*RXD        11 NC          # 3 GND         11*SS1S3     #
# 4 NC         12 DEN/R       # 4 NC         12 NC          # 4 GND         12*SS1S4     #
# 5 NC         13 INTA/R      # 5 NC         13 NC          # 5 GND         13*SS1S5     #
# 6 NC         14 VCC         # 6 NC         14 VCC         # 6 GND         14*SS1S6     #
# 7 GND                       # 7 GND                       # 7 GND         15*SS1S7     #
# 8*INTAINV                   # 8 NC                        # 8 GND         16*SS1S8     #
##########################################################################################
```

```
#U:17                          #E:11                                          #
#74C74 D-FF                    #MC1488 DRV                                    #
# 1 TXRDY        8 NC          # 1 VREG12       9 NC                          #
# 2 VCB         10 NC          # 2 NC          10 NC                          #
# 3 FFCLK       11 NC          # 3 NC          11 NC                          #
# 4 VCB         12 NC          # 4 TXD         12 NC                          #
# 5*FFOUT       13 NC          # 5 TXD         13 NC                          #
# 6 NC          14 VCC         # 6*XMIT232     14 VPOS12                      #
# 7 GND                        # 7 GND                                        #
# 8 NC                         # 8 NC                                         #
#################################################################################
#E:12                                                  #E:14                    #
#8251 USART                                            #RESET                   #
# 1 RD2          9 OUT0        17 GND        25 OUT0   # 1*VCB          9 NC   #
# 2 RD3         10 AIOWC/R     18 NC         26 VCC    # 2 VCC         10 GND  #
# 3 RXD         11 DEC02       19*TXD        27 RD0    # 3 GND         11 NC   #
# 4 GND         12 RA1         20 CLK0       28 RD1    # 4 NC          12 CBSQ #
# 5 RD4         13 IORC/R      21 RESET                # 5 VCC         13 NC   #
# 6 RD5         14 NC          22 NC                   # 6 INIT        14 INIT #
# 7 RD6         15*TXRDY       23 NC                   # 7 NC          15 INIT #
# 8 RD7         16 NC          24 NC                   # 8 NC          16 VCC  #
#################################################################################
#E:17                                                                           #
#8284 CLOCK                                                                     #
# 1 GND          9 GND         17 NC                                          #
# 2*CLK0        10*RESET       18 VCC                                         #
# 3 AEN         11 INIT                                                       #
# 4 RDY1        12 NC                                                         #
# 5*READY       13 GND                                                        #
# 6 RDY2        14 NC                                                         #
# 7 GND         15 NC                                                         #
# 8*CLK         16 NC                                                         #
#################################################################################
```

UNIT CONNECTIONS FOR DEMO NETWORK MOD PAGE 1

```
#E:18                                                                          #
#8088 TOP                                                                      #
# 1 GND          9 AD7          17*SINTR1      25*RQ/GT0      33 GND          #
# 2 AD14        10 AD6          18*SINTR2      26 S0          34 BHE          #
# 3 AD13        11 AD5          19 CLKX        27 S1          35 AD19         #
# 4 AD12        12 AD4          20 GND         28 S2          36 AD18         #
# 5 AD11        13 AD3          21 RESET       29 LOCK        37 AD17         #
# 6 AD10        14 AD2          22 READY       30 GND         38 AD16         #
# 7 AD9         15 AD1          23 CA          31 DRQ1        39 AD15         #
# 8 AD8         16 AD0          24 PAC         32 GND         40 VCC          #
################################################################################
#E:20                                    #E:21                                 #
#8288/B                                  #8282 LOW                             #
# 1 GND          9 NC           17 NC    # 1 AD0          9 GND          17*PA2 #
# 2 CLKX        10 GND          18 S2    # 2 AD1         10 GND          18*PA1 #
# 3 S1          11 NC           19 S0    # 3 AD2         11 ALE/R        19*PA0 #
# 4*DTR/B       12*AIOWC/R      20 VCC   # 4 AD3         12*PA7          20 VCC #
# 5*ALE/R       13*IORC/B                # 5 AD4         13*PA6                 #
# 6 GND         14*INTA/B                # 6 AD5         14*PA5                 #
# 7*MRDC/R      15 CEN/R                 # 7 AD6         15*PA4                 #
# 8*AMWC/B      16*DEN/B                 # 8 AD7         16*PA3                 #
################################################################################
#E:22                                                                          #
#8259A PIC                                                                      #
# 1 DEC03        9 AD2          17*INTR        25 NC                          #
# 2 AIOWC/B     10 AD1          18 SINTR1      26 INTA/R                      #
# 3 IORC/B      11 AD0          19 SINTR2      27 PA1                         #
# 4 AD7         12 NC           20 NC          28 VCC                         #
# 5 AD6         13 NC           21 NC                                         #
# 6 AD5         14 GND          22 NC                                         #
# 7 AD4         15 NC           23 NC                                         #
# 8 AD3         16 NC           24 NC                                         #
################################################################################
```

#E:19
#8288 TOP

# 1	GND	9	AD7	17*	SINTR1	25*	RQ/GT0	33	GND		
# 2	AD14	10	AD6	18*	SINTR2	26	S0	34	BHE		
# 3	AD13	11	AD5	19	CLKX	27	S1	35	AD19		
# 4	AD12	12	AD4	20	GND	28	S2	36	AD18		
# 5	AD11	13	AD3	21	RESET	29	LOCK	37	AD17		
# 6	AD10	14	AD2	22	READY	30	GND	38	AD16		
# 7	AD9	15	AD1	23	CA	31	DRQ1	39	AD15		
# 8	AD8	16	AD0	24	PA0	32	GND	40	VCC		

#E:20
#8288/R

# 1	GND	9	NC	17	NC		
# 2	CLKX	10	GND	18	S2		
# 3	S1	11	NC	19	S0		
# 4	*DTR/R	12*	AIOWC/R	20	VCC		
# 5	*ALE/R	13*	IORC/R				
# 6	GND	14*	INTA/R				
# 7	*MRDC/R	15	CEN/R				
# 8	*AMWC/R	16*	DEN/R				

#E:21
#8282 LOW

# 1	AD0	9	GND	17*	RA2		
# 2	AD1	10	GND	18*	RA1		
# 3	AD2	11	ALE/R	19*	RA0		
# 4	AD3	12*	RA7	20	VCC		
# 5	AD4	13*	RA6				
# 6	AD5	14*	RA5				
# 7	AD6	15*	RA4				
# 8	AD7	16*	RA3				

#E:22
#8259A PIC

# 1	DEC33	9	AD2	17*	INTR	25	NC
# 2	AIOWC/R	10	AD1	18	SINTR1	26	INTA/R
# 3	IORC/R	11	AD0	19	SINTR2	27	RA1
# 4	AD7	12	NC	20	NC	28	VCC
# 5	AD6	13	NC	21	NC		
# 6	AD5	14	GND	22	NC		
# 7	AD4	15	NC	23	NC		
# 8	AD3	16	NC	24	NC		

```
#P:1                                                                                    #
#SYS BUS                                                                                #
# 1*GND          9 NC          17*BUSY        25 NC          33 NC          41 NC      #
# 2 GND         10 NC          18 NC          26 NC          34 SA19        42 NC      #
# 3*VCC         11 GND         19 MRDC        27 BHEN        35 NC          43 SA14    #
# 4 VCC         12 GND         20 AMWC        28 SA16        36*INIT        44 SA15    #
# 5 VCC         13*BCLK        21 NC          29 CBRQ        37 NC          45 SA12    #
# 6 VCC         14 NC          22 NC          30 SA17        38 NC          46 SA13    #
# 7*VPOS12      15 NC          23*XACK        31 NC          39 NC          47 SA10    #
# 8 VPOS12      16 BPRO        24 NC          32 SA18        40 NC          48 SA11    #
#########################################################################################
#P:1                                                                    #P:2             #
#SYS BUS                                                               #PS232           #
 49 SA5         57 SA0         65 SD6         73 SD0         81 VCC     # 1 NC           #
 50 SA9         58 SA1         66 SD9         74 SD1         82 VCC     # 2*REC232       #
 51 SA6         59 SD14        67 SD8         75 GND         83 VCC     # 3 XMIT232      #
 52 SA7         60 SD15        68 SD7         76 GND         84 VCC     # 4 NC           #
 53 SA4         61 SD12        69 SD4         77 NC          85 GND     # 5 NC           #
 54 SA5         62 SD13        70 SD5         78 NC          86 GND     # 6 NC           #
 55 SA2         63 SD10        71 SD2         79*VNEG12                 # 7 GND          #
 56 SA3         64 SD11        72 SD3         80 VNEG12                                   #
#########################################################################################
```

SIGNAL	FANOUT	SOURCE	SINKS
AD0	8	A:19/16	A:15/1 A:18/1 B:11/8 B:14/4 B:21/1 E:18/16 E:21/1 E:22/11
AD1	7	A:19/15	A:15/2 A:18/2 B:11/7 B:21/2 E:18/15 E:21/2 E:22/10
AD10	6	A:19/6	A:14/3 A:16/3 A:17/3 A:21/3 C:21/3 E:18/6
AD11	6	A:19/5	A:14/4 A:16/4 A:17/4 A:21/4 C:21/4 E:18/5
AD12	6	A:19/4	A:14/5 A:16/5 A:17/5 A:21/5 C:21/5 E:18/4
AD13	6	A:19/3	A:14/6 A:16/6 A:17/6 A:21/6 C:21/6 E:18/3
AD14	6	A:19/2	A:14/7 A:16/7 A:17/7 A:21/7 C:21/7 E:18/2
AD15	6	A:19/39	A:14/8 A:16/8 A:17/8 A:21/8 C:21/8 E:18/39
AD16	3	A:19/35	A:13/1 C:20/5 E:18/38
AD17	3	A:19/37	A:13/2 C:20/6 E:18/37
AD18	3	A:19/36	A:13/3 C:20/7 E:18/36
AD19	3	A:19/35	A:13/4 C:20/8 E:16/35

SIGNAL	FANOUT	SOURCE	SINKS
AD2	7	A:19/14	A:15/3 A:18/3 B:11/6 B:21/3 E:18/14 E:21/3 E:22/9
AD3	7	A:19/13	A:15/4 A:18/4 B:11/5 B:21/4 E:18/13 E:21/4 E:22/8
AD4	7	A:19/12	A:15/5 A:18/5 B:11/4 B:21/5 E:18/12 E:21/5 E:22/7
AD5	7	A:19/11	A:15/6 A:18/6 B:11/3 B:21/6 E:18/11 E:21/6 E:22/6
AD6	7	A:19/10	A:15/7 A:18/7 B:11/2 B:21/7 E:18/10 E:21/7 E:22/5
AD7	7	A:19/9	A:15/8 A:18/8 B:11/1 B:21/8 E:18/9 E:21/8 E:22/4
AD8	6	A:19/8	A:14/1 A:16/1 A:17/1 A:21/1 C:21/1 E:18/8
AD9	6	A:19/7	A:14/2 A:16/2 A:17/2 A:21/2 C:21/2 E:18/7
AE	5	A:12/13	A:11/6 A:13/9 A:14/9 A:15/9 E:17/3
AG B	1	C:14/5	B:15/5
ALEMC/	6	E:20/12	B:11/23 B:14/2 B:16/10 C:18/1 E:12/10 E:22/2
ALE	3	A:11/5	A:13/11 A:14/11 A:15/11

SIGNAL	FANOUT	SOURCE	SINKS
ALE/P	3	E:20/5	C:20/11 C:21/11 E:21/11
ALTB	1	C:15/7	B:15/4
AMWC	1	A:11/8	P:1/20
AMWC/R	3	E:20/8	A:22/27 B:22/27 C:18/4
BCLK	1	P:1/13	A:12/5
BHE	3	A:15/34	B:14/3 C:20/1 E:18/34
BHE/4B	1	B:14/6	A:13/8
BHEL	1	C:20/19	B:18/1
BHEN	2	A:13/12	B:15/10 P:1/27
BP-C	1	A:12/8	P:1/16
BUSY	1	P:1/17	A:12/11
CA	1	B:13/10	C:18/23
CAX	1	B:16/6	B:13/11
CB Q	2	A:12/12	C:14/12 P:1/29
CF	3	C:14/11	A:11/15 A:12/3 B:17/14

SIGNAL	FANOUT	SOURCE	SINKS
CEN/F	3	B:15/6	C:18/12,13 E:20/15
CLK	2	E:17/8	C:16/9,10
CLK0	2	E:17/2	B:11/9 E:12/20
CLKX	3	C:16/8	A:19/19 E:18/19 E:20/2
CLKY	2	C:16/7	A:11/2 A:12/17
DDEN/	1	C:24/11	B:18/13
DEC00	2	B:17/1	B:18/2,4
DEC01	2	B:17/4	B:11/21 C:18/10
DEC02	2	B:17/2	B:14/1 E:12/11
DEC03	2	B:17/3	C:18/9 E:22/1
DEC04	1	B:17/5	B:18/9
DE	2	A:11/15	B:15/12,13
DEN/	1	C:20/16	C:24/12
DFINV	2	B:15/11	B:16/5,9
DFQ1	1	B:14/8	E:18/31
DTF	3	A:11/4	A:16/11 A:17/11 A:18/11

SIGNAL	FANOUT	SOURCE	SINKS
DTP/P	2	E:20/4	A:21/11 B:21/11
FFCLK	1	B:14/12	D:17/3
FQOUT	1	D:1//5	B:14/10
GND	82	P:1/1	A:11/1,10 A:12/9,10,14 A:13/10 A:14/10 A:15/10 A:16/10 A:17/10 A:18/10 A:19/1,17,20,33 A:21/10 A:22/14 B:11/11,12 B:13/7 B:14/5,7,11 B:15/7 B:16/7 B:17/8,15 B:18/7 B:21/10 B:22/14 C:14/2,3,4,8 C:15/2 C:15/3,4,8 C:18/7 C:20/9 C:20/10 C:21/9,10 C:24/7 D:11/7 D:14/1,2,3,4,5,6 D:14/7,8 D:17/7 E:11/7 E:12/4,17 E:14/3,10 E:17/1 E:17/7,9,13 E:18/1,20,30 E:18/32,33 E:20/1,6,10 E:21/9,10 E:22/14 P:1/2 P:1/11,12,75,76,85,86 P:2/7
HI AM	1	B:1[illegible]/3	A:22/20
INIT	5	P:1/36	A:12/6 E:14/6,14,15 E:17/11
INTA/	4	C:2[illegible]/14	C:24/9,10,13 E:22/26
INTAINV	1	C:24/8	B:16/12
IN R	1	E:22/17	A:19/18
IOPC/	4	E:20/13	B:11/22 C:19/2 E:12/13 E:22/3
L1	3	B:15/8	B:15/1,2 B:16/4

SIGNAL	FANOUT	SOURCE	SINKS
L2	1	B:15/3	B:16/10
LOCK	2	A:13/29	A:12/16 E:18/29
LOWRAM	1	B:18/6	B:22/20
MPDC	1	A:11/7	P:1/19
MPDC/P	3	E:20/7	A:22/22 B:22/22 C:18/5
OE/6BH	1	B:1E/6	A:16/9
OE/6BC	2	B:16/8	A:17/9 A:18/9
OEDIS	1	C:16/8	B:18/12
OEI	2	B:15/11	A:21/9 B:21/9
OUT0	2	B:11/10	C:12/9,25
PAC	3	E:21/19	B:15/9 B:18/5 E:18/24
PA	6	E:21/18	A:22/10 B:11/19 B:14/13 B:22/10 E:12/12 E:22/27
PA10	2	C:21/17	A:22/24 B:22/24
PA11	2	C:21/16	A:22/21 B:22/21
PA12	3	C:21/15	A:22/23 B:17/10 B:22/23

SIGNAL LIST FOR DEMO NETWORK MCD PAGE 7

SIGNAL	FANOUT	SOURCE	SINKS
FA13	3	C:21/14	A:22/2 B:17/11 B:22/2
FA14	1	C:21/13	B:17/12
FA15	1	C:21/12	B:17/13
FA16	2	C:20/15	C:14/10 C:15/10
FA17	2	C:20/14	C:14/12 C:15/12
FA18	2	C:20/13	C:14/13 C:15/13
FA19	2	C:20/12	C:14/15 C:15/15
FA2	3	E:21/17	A:22/9 B:11/20 B:22/9
FA3	2	E:21/16	A:22/8 B:22/8
FA4	2	E:21/15	A:22/7 B:22/7
FA5	2	E:21/14	A:22/6 B:22/6
FA6	2	E:21/13	A:22/5 B:22/5
FA7	2	E:21/12	A:22/4 B:22/4
FA8	2	C:21/19	A:22/3 B:22/3
FA9	2	C:21/18	A:22/25 B:22/25
FD0	2	B:21/19	B:22/11 E:12/27

SIGNAL	FANOUT	SOURCE	SINKS	
FD1	2	B:21/18	B:22/12	E:12/28
FD10	1	A:21/17	A:22/13	
FD11	1	A:21/16	A:22/15	
FD12	1	A:21/15	A:22/16	
FD13	1	A:21/14	A:22/17	
FD14	1	A:21/13	A:22/18	
FD15	1	A:21/12	A:22/19	
FD2	2	B:21/17	B:22/13	E:12/1
FD3	2	B:21/16	B:22/15	E:12/2
FD4	2	B:21/15	B:22/16	E:12/5
FD5	2	B:21/14	B:22/17	E:12/6
FD6	2	B:21/13	B:22/18	E:12/7
FD7	2	B:21/12	B:22/19	E:12/8
FDC	1	A:21/19	A:22/11	
RDS	1	A:21/18	A:22/12	
RDY1	1	B:13/6	E:17/4	
RDY2	1	B:16/11	E:17/6	

SIGNAL	FANOUT	SOURCE	SINKS
RDYIO	1	C:18/3	B:16/1
RDYM	1	C:18/6	B:16/2
RDYX	1	B:16/3	B:16/13
READY	2	E:17/5	A:19/22 E:18/22
REC232	1	P:2/2	D:11/1
RESET	3	E:17/10	A:19/21 E:12/21 E:18/21
RQ/GT0	1	E:18/25	A:19/31
RXD	1	D:11/3	E:12/3
S0	4	A:19/26	A:11/19 A:12/18 E:18/26 E:20/19
S1	4	A:19/27	A:11/3 A:12/19 E:18/27 E:20/3
S2	4	A:19/28	A:11/18 A:12/1 E:18/28 E:20/18
SA0	1	A:15/19	P:1/57
SA1	1	A:15/18	P:1/58
SA10	1	A:14/17	P:1/47
SA11	1	A:14/16	P:1/48
SA12	1	A:14/15	P:1/45

SIGNAL	FANOUT	SOURCE	SINKS
SA13	1	A:14/14	P:1/46
SA14	1	A:14/13	P:1/43
SA15	1	A:14/12	P:1/44
SA16	1	A:13/19	P:1/28
SA17	1	A:13/18	P:1/30
SA18	1	A:13/17	P:1/32
SA19	1	A:13/16	P:1/34
SA2	1	A:15/17	P:1/55
SA3	1	A:15/16	P:1/56
SA4	1	A:15/15	P:1/53
SA5	1	A:15/14	P:1/54
SA6	1	A:15/13	P:1/51
SA7	1	A:15/12	P:1/52
SA8	1	A:14/19	P:1/49
SA9	1	A:14/18	P:1/50
SD0	2	A:16/19	A:18/19 P:1/73
SD1	2	A:16/18	A:18/18 P:1/74

SIGNAL	FANOUT	SOURCE	SINKS	
SD10	1	A:17/17	P:1/63	
SD11	1	A:17/16	P:1/64	
SD12	1	A:17/15	P:1/61	
SD13	1	A:17/14	P:1/62	
SD14	1	A:17/13	P:1/59	
SD15	1	A:17/12	P:1/60	
SD2	2	A:16/17	A:18/17	P:1/71
SD3	2	A:16/16	A:18/16	P:1/72
SD4	2	A:16/15	A:18/15	P:1/69
SD5	2	A:16/14	A:18/14	P:1/70
SD6	2	A:16/13	A:18/13	P:1/67
SD7	2	A:16/12	A:18/12	P:1/68
SD8	1	A:17/19	P:1/65	
SD9	1	A:17/18	P:1/66	
SINTR1	1	E:18/17	E:22/18	
SINTR2	1	E:18/18	E:22/19	
SS1S1	1	D:14/9	C:14/9	

SIGNAL	FANOUT	SOURCE	SINKS
SS1S2	1	D:14/10	C:14/11
SS1S3	1	D:14/11	C:14/14
SS1S4	1	D:14/12	C:14/1
SS1S5	1	D:14/13	C:15/9
SS1S6	1	D:14/14	C:15/11
SS1S7	1	D:14/15	C:15/14
SS1S8	1	D:14/16	C:15/1
TXD	2	E:12/19	E:11/4,5
TXINV	1	B:13/12	B:14/9
TXRDY	2	E:12/15	B:13/13 D:17/1
VCB	2	E:14/1	D:17/2,4
VCC	47	P:1/3	A:11/20 A:12/2,4,15,20 A:13/20 A:14/20 A:15/20 A:16/20 A:17/20 A:18/20 A:19/40 A:21/20 A:22/28 B:11/24 B:13/14 B:14/14 B:15/14 B:16/14 B:17/16 B:18/14 B:21/20 B:22/28 C:14/16 C:15/16 C:18/14 C:20/20 C:21/20 C:24/14 D:11/14 D:17/14 E:12/26 E:14/2,5,16 E:17/18 E:18/40 E:20/20 E:21/20 E:22/28 P:1/4,5,6,81,82,83,84
VME612	2	P:1/79	E:11/1 P:1/80

SIGNAL	FANOUT	SOURCE	SINKS
VPCS12	2	P:1/7	E:11/14 P:1/8
XACK	1	P:1/23	R:13/5
XMIT232	1	E:11/6	P:2/3

A:11/2	A:12/17	CLKY
A:11/4	A:16/11	DTR
A:11/5	A:13/11	ALE
A:11/6	A:12/13	AEN
A:11/7	P:1/19	MRDC
A:11/8	P:1/20	AMWC
A:11/15	A:12/3	CEN
A:12/1	A:19/28	S2
A:12/5	P:1/13	BCLK
A:12/6	E:14/15	INIT
A:12/8	P:1/16	BPRO
A:12/11	P:1/17	BUSY
A:12/18	A:19/26	S0
A:12/19	A:19/27	S1
A:13/1	A:19/38	AD16
A:13/2	A:19/37	AD17
A:13/3	A:19/36	AD18
A:13/4	A:19/35	AD19
A:13/8	B:14/6	BHE/4B
A:13/9	A:14/9	AEN
A:13/16	P:1/34	SA19
A:13/17	P:1/32	SA18
A:13/18	P:1/30	SA17
A:13/19	P:1/28	SA16
A:14/1	A:16/1	AD8
A:14/2	A:16/2	AD9
A:14/3	A:16/3	AD10
A:14/4	A:16/4	AD11
A:14/5	A:16/5	AD12
A:14/6	A:16/6	AD13
A:14/7	A:16/7	AD14
A:14/8	A:16/8	AD15
A:14/11	A:15/11	ALE
A:14/12	P:1/49	SA15
A:14/13	P:1/43	SA14
A:14/14	P:1/46	SA13
A:14/15	P:1/45	SA12
A:14/16	P:1/48	SA11
A:14/17	P:1/47	SA10
A:14/18	P:1/50	SA9

A:14/15	P:1/49	SA8
A:15/1	A:18/1	AD0
A:15/2	B:11/7	AD1
A:15/3	B:11/6	AD2
A:15/4	B:11/5	AD3
A:15/5	B:11/4	AD4
A:15/6	B:11/3	AD5
A:15/7	B:11/2	AD6
A:15/8	B:11/1	AD7
A:15/9	E:17/3	AEN
A:15/12	P:1/52	SA7
A:15/13	P:1/51	SA6
A:15/14	P:1/54	SA5
A:15/15	P:1/53	SA4
A:15/16	P:1/56	SA3
A:15/17	P:1/55	SA2
A:15/18	P:1/58	SA1
A:15/19	P:1/57	SA0
A:16/9	B:16/6	OE/6BB
A:17/1	A:19/8	AD8
A:17/2	A:19/7	AD9
A:17/3	A:19/6	AD10
A:17/4	A:19/5	AD11
A:17/5	A:19/4	AD12
A:17/6	A:19/3	AD13
A:17/7	A:19/2	AD14
A:17/8	A:19/39	AD15
A:17/9	A:18/9	OE/6BC
A:17/11	A:18/11	DIR
A:17/12	P:1/60	SD15
A:17/13	P:1/59	SD14
A:17/14	P:1/62	SD13
A:17/15	P:1/61	SD12
A:17/16	P:1/64	SD11
A:17/17	P:1/63	SD10
A:17/18	P:1/66	SD9
A:17/19	P:1/65	SD8
A:18/2	A:19/15	AD1
A:18/3	A:19/14	AD2
A:18/4	A:19/13	AD3

A:18/5	A:19/12	AD4
A:18/6	A:19/11	AD5
A:18/7	A:19/10	AD6
A:18/8	A:19/9	AD7
A:18/12	P:1/68	SD7
A:18/13	P:1/67	SD6
A:18/14	P:1/70	SD5
A:18/15	P:1/69	SD4
A:18/16	P:1/72	SD3
A:18/17	P:1/71	SD2
A:18/18	P:1/74	SD1
A:18/19	P:1/73	SD0
A:19/16	E:18/16	AD0
A:19/18	E:22/17	INTR
A:19/19	E:20/2	CLKX
A:19/21	E:18/21	RESET
A:19/22	E:18/22	READY
A:19/29	E:18/29	LOCK
A:19/31	E:18/25	RQ/GT0
A:19/34	B:14/3	BHE
A:21/9	B:21/9	OER
A:21/11	B:21/11	DTR/R
A:21/12	A:22/19	RD15
A:21/13	A:22/18	RD14
A:21/14	A:22/17	RD13
A:21/15	A:22/16	RD12
A:21/16	A:22/15	RD11
A:21/17	A:22/13	RD10
A:21/18	A:22/12	RD9
A:21/19	A:22/11	RD8
A:22/2	B:22/2	RA13
A:22/3	B:22/3	RA8
A:22/4	B:22/4	RA7
A:22/5	B:22/5	RA6
A:22/6	B:22/6	RA5
A:22/7	B:22/7	RA4
A:22/8	B:22/8	RA3
A:22/9	B:11/20	RA2
A:22/10	B:14/13	RA1
A:22/20	B:18/3	HIRAM
A:22/21	B:22/21	RA11
A:22/22	B:22/22	MRDC/R

A:22/23	B:22/23	PA12
A:22/24	B:22/24	RA10
A:22/25	B:22/25	RA9
A:22/27	B:22/27	AMWC/R
B:11/8	B:14/4	AD0
B:11/9	E:12/20	CLK0
B:11/19	E:12/12	RA1
B:11/22	C:18/2	IORC/R
B:11/23	B:14/2	AIOWC/R
B:13/5	P:1/23	XACK
B:13/6	E:17/4	RDY1
B:13/10	E:18/23	CA
B:13/11	B:18/8	CAX
B:13/12	B:14/9	TXINV
B:13/13	D:17/1	TXRDY
B:14/1	E:12/11	DECO2
B:14/8	E:18/31	DRQ1
B:14/10	D:17/5	FQOUT
B:14/12	D:17/3	FFCLK
B:15/1	B:15/8	L1
B:15/2	B:16/4	L1
B:15/3	B:16/10	L2
B:15/4	C:15/7	ALTB
B:15/5	C:14/5	AGTB
B:15/6	C:18/12	CEN/R
B:15/9	B:18/5	RA0
B:15/10	P:1/27	BHEN
B:15/12	B:15/13	DEN
B:16/1	C:18/3	RDYIO
B:16/2	C:18/6	RDYM
B:16/3	B:16/13	RDYX
B:16/5	B:16/9	DENINV
B:16/11	E:17/6	RDY2
B:16/12	C:24/8	INTAINV
B:17/4	C:18/10	DECO1
B:17/5	B:18/9	DECO4

B:17/10	C:21/15	RA12
B:17/11	C:21/14	RA13
B:17/12	C:21/13	RA14
B:17/13	C:21/12	RA15
B:17/14	C:18/11	CEN
B:18/1	C:20/19	BHEL
B:18/2	B:18/4	DECOO
B:18/6	B:22/20	LOWRAM
B:18/10	C:18/1	AIOWC/R
B:18/12	C:18/8	OEDIS
B:18/13	C:24/11	DDEN/R
B:21/2	E:22/10	AD1
B:21/3	E:22/9	AD2
B:21/4	E:22/8	AD3
B:21/5	E:22/7	AD4
B:21/6	E:22/6	AD5
B:21/7	E:22/5	AD6
B:21/8	E:22/4	AD7
B:21/12	B:22/19	RD7
B:21/13	B:22/18	RD6
B:21/14	B:22/17	RD5
B:21/15	B:22/16	RD4
B:21/16	B:22/15	RD3
B:21/17	B:22/13	RD2
B:21/18	B:22/12	RD1
B:21/19	B:22/11	RD0
B:22/9	E:21/17	RA2
B:22/10	E:22/27	RA1
C:14/1	D:14/12	SS1S4
C:14/9	D:14/9	SS1S1
C:14/10	C:15/10	RA16
C:14/11	D:14/10	SS1S2
C:14/12	C:15/12	RA17
C:14/13	C:15/13	RA18
C:14/14	D:14/11	SS1S3
C:14/15	C:15/15	RA19
C:15/1	D:14/16	SS1S8
C:15/9	D:14/13	SS1S5

C:15/11	D:14/14	SS1S6
C:15/14	D:14/15	SS1S7
C:16/4	E:16/19	CLKX
C:16/9	C:16/10	CLK
C:18/4	E:20/8	AMWC/R
C:18/5	E:20/7	MRDC/R
C:18/9	E:22/1	DEC03
C:18/13	E:20/15	CEN/R
C:20/1	E:16/34	BHE
C:20/5	E:18/38	AD16
C:20/6	E:18/37	AD17
C:20/7	E:18/36	AD18
C:20/8	E:18/35	AD19
C:20/11	C:21/11	ALE/R
C:21/1	E:16/8	AD8
C:21/2	E:18/7	AD9
C:21/3	E:18/6	AD10
C:21/4	E:18/5	AD11
C:21/5	E:18/4	AD12
C:21/6	E:18/3	AD13
C:21/7	E:18/2	AD14
C:21/8	E:18/39	AD15
C:24/9	E:22/26	INTA/R
C:24/10	C:24/13	INTA/R
C:24/12	E:20/16	DEN/R
D:11/1	P:2/2	REC232
D:11/3	E:12/3	RXD
D:17/2	D:17/4	VCB
E:11/1	P:1/80	VNEG12
E:11/4	E:11/5	TXD
E:11/6	P:2/3	XMIT232
E:11/14	P:1/8	VPOS12

E:12/9	E:12/25	OUT0
E:12/21	E:11/10	RESET
E:14/6	E:14/14	INIT
E:14/12	P:1/29	CBRQ
E:17/11	P:1/36	INIT
E:18/9	E:21/8	AD7
E:18/10	E:21/7	AD6
E:18/11	E:21/6	AD5
E:18/12	E:21/5	AD4
E:18/13	E:21/4	AD3
E:18/14	E:21/3	AD2
E:18/15	E:21/2	AD1
E:18/17	E:22/16	SINTR1
E:18/18	E:22/19	SINTR2
E:18/24	E:21/19	RA0
E:18/26	E:20/19	S0
E:18/27	E:20/3	S1
E:18/28	E:20/18	S2
E:20/5	E:21/11	ALE/R
E:20/12	E:22/2	AIOWC/R
E:20/13	E:22/3	IORC/R
E:21/1	E:22/11	ADC

A:11/3	A:12/19	S1
A:11/16	B:15/12	DEN
A:11/18	A:12/1	S2
A:11/19	A:12/16	S0
A:12/3	B:17/14	CEN
A:12/12	E:14/12	CBRQ
A:12/13	A:13/9	AEN
A:12/16	A:19/29	LOCK
A:12/17	C:16/7	CLKY
A:13/11	A:14/11	ALE
A:13/12	B:15/10	BHEN
A:14/9	A:15/9	AEN
A:15/1	B:14/4	AD0
A:15/2	A:18/2	AD1
A:15/3	A:18/3	AD2
A:15/4	A:18/4	AD3
A:15/5	A:18/5	AD4
A:15/6	A:18/6	AD5
A:15/7	A:16/7	AD6
A:15/8	A:18/8	AD7
A:16/1	A:17/1	AD8
A:16/2	A:17/2	AD9
A:16/3	A:17/3	AD10
A:16/4	A:17/4	AD11
A:16/5	A:17/5	AD12
A:16/6	A:17/6	AD13
A:16/7	A:17/7	AD14
A:16/8	A:17/8	AD15
A:16/11	A:17/11	DTR
A:16/12	A:18/12	SD7
A:16/13	A:18/13	SD6
A:16/14	A:18/14	SD5
A:16/15	A:18/15	SD4
A:16/16	A:18/16	SD3
A:16/17	A:18/17	SD2
A:16/18	A:18/18	SD1

A:16/15	A:18/19	SD0
A:17/9	B:16/8	OE/68C
A:16/1	A:19/16	AD0
A:19/2	E:18/2	AD14
A:19/3	E:18/3	AD13
A:19/4	E:18/4	AD12
A:19/5	E:18/5	AD11
A:19/6	E:18/6	AD10
A:19/7	E:18/7	AD9
A:19/8	E:18/8	AD8
A:19/9	E:18/9	AD7
A:19/10	E:18/10	AD6
A:19/11	E:19/11	AD5
A:19/12	E:18/12	AD4
A:19/13	E:18/13	AD3
A:19/14	E:18/14	AD2
A:19/15	E:18/15	AD1
A:19/19	E:18/19	CLKX
A:19/26	E:18/26	S0
A:19/27	E:18/27	S1
A:19/28	E:19/28	S2
A:19/34	E:18/34	BHE
A:19/35	E:18/35	AD19
A:19/36	E:18/36	AD18
A:19/37	E:18/37	AD17
A:19/38	E:18/38	AD16
A:19/39	E:18/39	AD15
A:21/1	C:21/1	AD8
A:21/2	C:21/2	AD9
A:21/3	C:21/3	AD10
A:21/4	C:21/4	AD11
A:21/5	C:21/5	AD12
A:21/6	C:21/6	AD13
A:21/7	C:21/7	AD14
A:21/8	C:21/8	AD15
A:22/9	B:22/9	RA2
A:22/10	B:22/10	RA1
B:11/13	E:12/9	OUT0

B:11/10	B:14/13	RA1
B:11/21	B:17/4	DEC01
B:11/22	E:12/13	IORC/R
B:11/23	E:12/10	AIOWC/R
B:13/13	E:12/15	TXRDY
B:14/1	B:17/2	DEC02
B:14/2	B:18/10	AIOWC/R
B:15/1	B:15/2	L1
B:15/11	B:16/5	DENINV
B:17/1	B:16/2	DEC00
B:17/3	C:18/9	DEC03
B:18/5	E:18/24	RA0
B:18/11	B:21/9	OER
B:21/1	E:22/11	AD0
B:21/11	E:20/4	DTR/R
B:21/12	E:12/8	RD7
B:21/13	E:12/7	RD6
B:21/14	E:12/6	RD5
B:21/15	E:12/5	RD4
B:21/16	E:12/2	RD3
B:21/17	E:12/1	RD2
B:21/18	E:12/26	RD1
B:21/19	E:12/27	RD0
B:22/2	C:21/14	RA13
B:22/3	C:21/15	RA8
B:22/4	E:21/12	RA7
B:22/5	E:21/13	RA6
B:22/6	E:21/14	RA5
B:22/7	E:21/15	RA4
B:22/8	E:21/16	RA3
B:22/21	C:21/16	RA11
B:22/22	E:20/7	MRDC/R
B:22/23	C:21/19	RA12
B:22/24	C:21/17	RA10

B:22/25	C:21/1F	RA9
B:22/27	E:20/6	AMWC/R
C:15/11	C:20/15	RA16
C:15/12	C:20/14	RA17
C:15/13	C:20/13	RA18
C:15/15	C:20/12	RA19
C:16/9	E:17/3	CLK
C:18/1	E:20/12	AIOWC/R
C:18/2	E:20/13	IORC/R
C:18/12	C:18/13	CEN/R
C:21/11	E:21/11	ALE/R
C:24/9	C:24/10	INTA/R
D:17/2	E:14/1	VCB
E:11/4	E:12/19	TXD
E:12/20	E:17/2	CLKO
E:14/9	E:17/11	INIT
E:14/14	E:14/15	INIT
E:17/5	E:18/22	READY
E:17/10	E:18/21	RESET
E:18/18	E:21/1	AD0
E:20/14	E:22/26	INTA/R
E:21/2	E:22/10	AD1

E:21/3	E:22/9	AD2
E:21/4	E:22/8	AD3
E:21/5	E:22/7	AD4
E:21/6	E:22/6	AD5
E:21/7	E:22/5	AD6
E:21/8	E:22/4	AD7
E:21/16	E:22/27	RA1

P:1/7	P:1/8	VPOS12
P:1/79	P:1/80	VNEG12

```
  CSB   NOS/BE L564D        L564-CMR3  11/15/82
12.29.23.DEPAF5G  FROM CSA/AF
12.29.24.IP  00000766 WORDS - FILE INPUT  , DC 04
12.29.24.DEP,T15,CM77000,CU20,,T820665,PALMER,435
12.29.24.1,55533
12.29.27.ATTACH,WLIST,ID=F770050,SN=ASDEN,CY=1.
12.29.27.PFN IC
12.29.27.WLIST
12.29.27.LIBRARY,COBOL.
12.29.27.COPY,INPUT,WLDATA.
12.29.29.      NON-FATAL LOADER ERRORS -
12.29.29.NON-EXISTENT LIBRARY GIVEN -  COBOL
12.29.31.REWIND,WLDATA,WLIST.
12.29.31.      NON-FATAL LOADER ERRORS -
12.29.31.NON-EXISTENT LIBRARY GIVEN -  COBOL
12.29.33.WLIST.
12.29.36.      NON-FATAL LOADER ERRORS -
12.29.36.NON-EXISTENT LIBRARY GIVEN -  COBOL
12.30.17.      STOP
12.30.17.      067100 MAXIMUM EXECUTION FL.
12.30.17.      11.438 CP SECONDS EXECUTION TIME.
12.30.17.REWIND,WLDATA,WLIST.
12.30.17.      NON-FATAL LOADER ERRORS -
12.30.17.NON-EXISTENT LIBRARY GIVEN -  COBOL
12.30.17.CP  00008064 WORDS - FILE OUTPUT , DC 40
12.30.17.MS   10244  WORDS  (   4012  MAX USED)
12.30.17.CPA      12.319 SEC.        6.167 ADJ.
12.30.18.IO       19.163 SEC.       10.846 ADJ.
12.30.18.CM      600.961 KWS.        4.979 ADJ.
12.30.18.CRUS                       21.693
12.30.18.PP       17.472 SEC.     DATE 12/04/82
12.30.18.EJ  END OF JOB, AF  T820665.
```

Appendix C

Pin-out Diagrams of Network Module Components

This appendix contains the pin-out diagrams for the Intel integrated circuit chips used in the Network Module. They are included here to serve as a quick reference to their signal connections. For more detailed information, the respective data sheets should be consulted. The pin-out diagrams are shown in Figure C1.

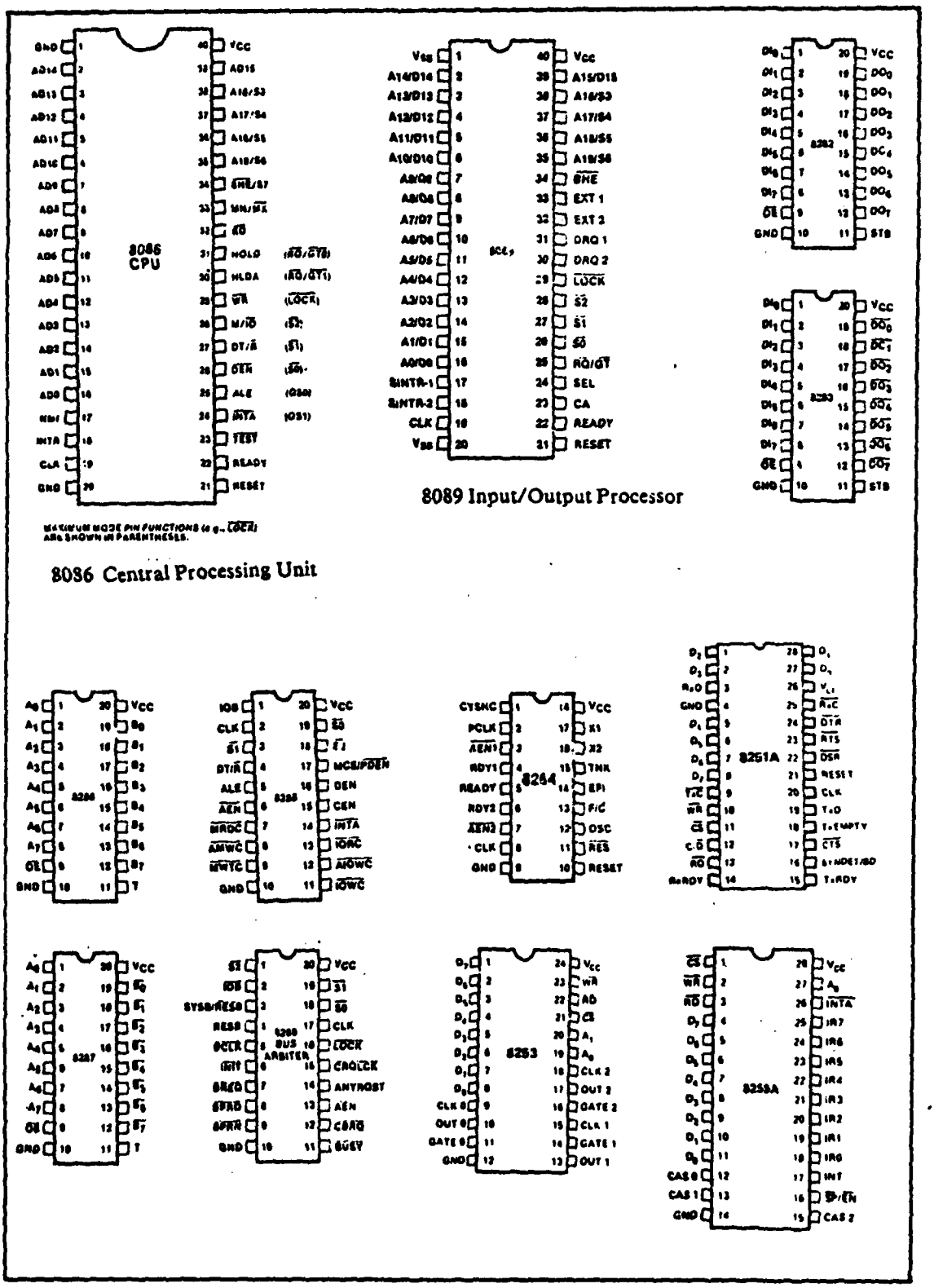


Figure C-1. Integrated Circuit Pin-out Diagrams

Appendix D

PROM Programming Procedure

A 74S288 Programmable Read Only Memory (PROM) was used as an address decoder for the network module. There was not a PROM programmer available, so the PROM was programmed using a Wavetex Model 186 signal generator, two dc power supplies, a voltmeter, and an 8016A word generator. I followed the programming procedures specified by the manufacturer (Ref 36:185.

The PROM was programmed as follows: The Wavetex was connected to the PROM to provide the Vcc voltage. Its DC Bias was set at 5v, so I was able to obtain approximately 9.5v out. (The manufacturer specifies a minimum of 10v for Vcc during programming, but 9.5v was sufficent). The Wavetex pulse repetition rate was set at 8ms, with a pulse width of approximately 1.4 ms. This was well below the 35% duty cycle specified. The word generator was used to sync the Wavetex and to supply the chip select ($\overline{CS}$) signal to the PROM. Each bit had to be programmed individually, and all the outputs which were not being programmed were tied thru 3.8K ohm resistors to the +5v supplied by one of the power supplies. The other power supply was used to supply the .28vdc programming pulse to the PROM output being programmed.

With the above setup, the PROM was programmed for the desired outputs by using shorting wires to apply the desired addresses to the PROM address pins. Then touching the +

.28v lead to the output needing programmed. The PROM has all of its outputs low. Since I wanted an active low $\overline{CS}$ signal, all bit positions had to have the .28v applied, except for the one at the desired address input.

Appendix E

Physical Layout of Network Module Components

This appendix contains the component layout of the components installed on the Network Module's wire-wrap board. The components are numbered by row and column. The physical layout is shown in Figure E1. Looking at the component side (side shown), and with the Multibus connector (the 86 pin connector), on top, the component in the upper right hand corner is designated A11. The one to its immediate left is A12. The row below the first is B, then C, D, and then E. The components which span over two rows, such as the 8089 (E18), are only given one designation.

The components are arranged somewhat into functional groups. The components A11-A18 are system bus interface chips, except that A12 is the 8289 bus arbiter. Components A21, B21, C20, C21, E20, and E21 are the resident bus interface components. The dashed lines at positions E12 and E24 indicate where the Signetics 2652's are to be installed.

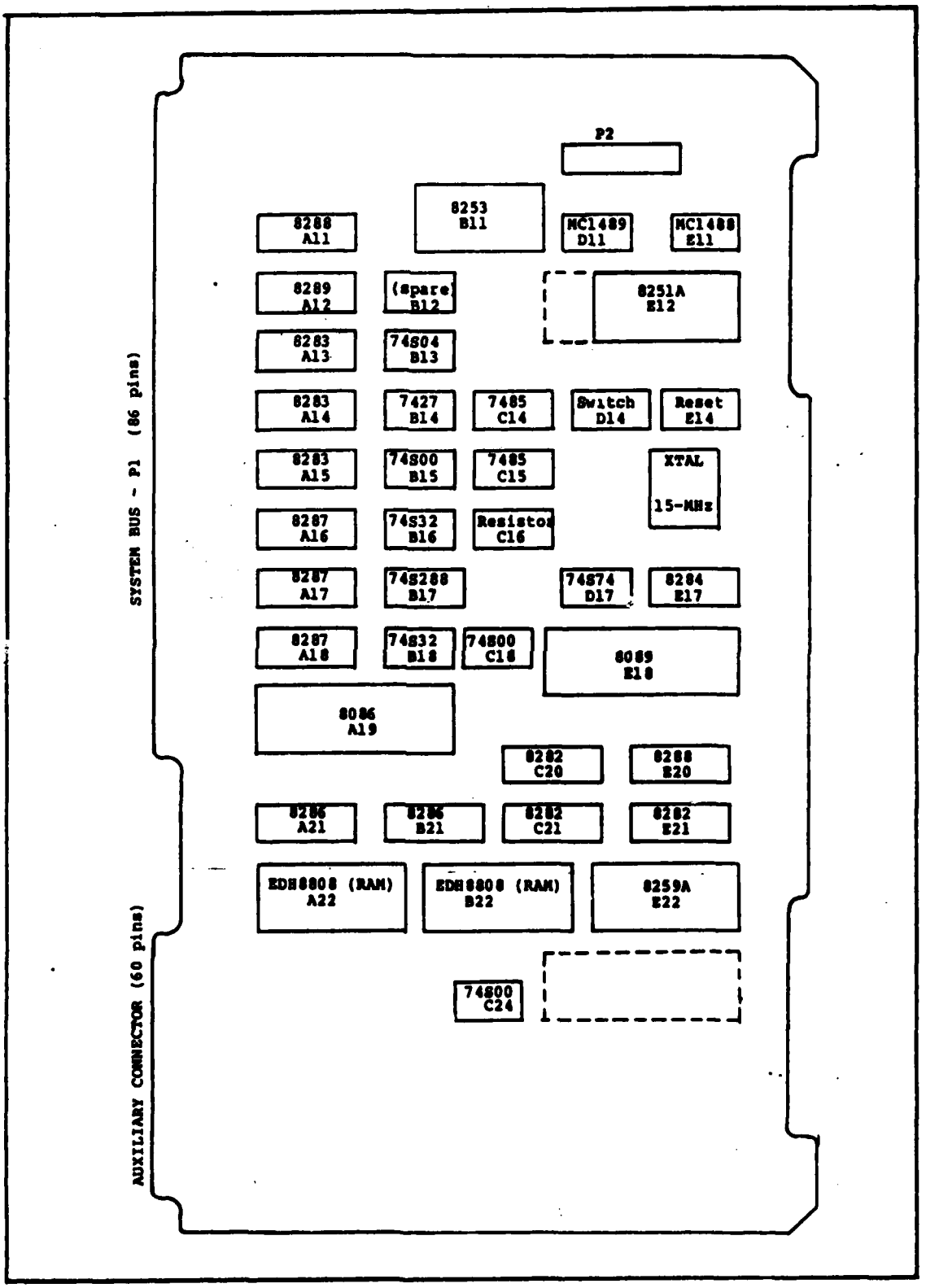


Figure E-1. Physical Layout of Network Module Components

Appendix F

Schematic Diagram of Demonstration Network Module

This appendix contains the schematic diagram of the demonstration network module. All connections are the same as on the final network module, except that the USART (E12) will not be included and the two 2652-1 Multiprotocol Communication Controllers will be connected. The schematic is divided into three seperate sheets and it is shown in Figures F1, F2, and F3. There are .068 microfarad bypass capacitors connected to each integrated circuits Vcc pin, which are not shown on the schematic.

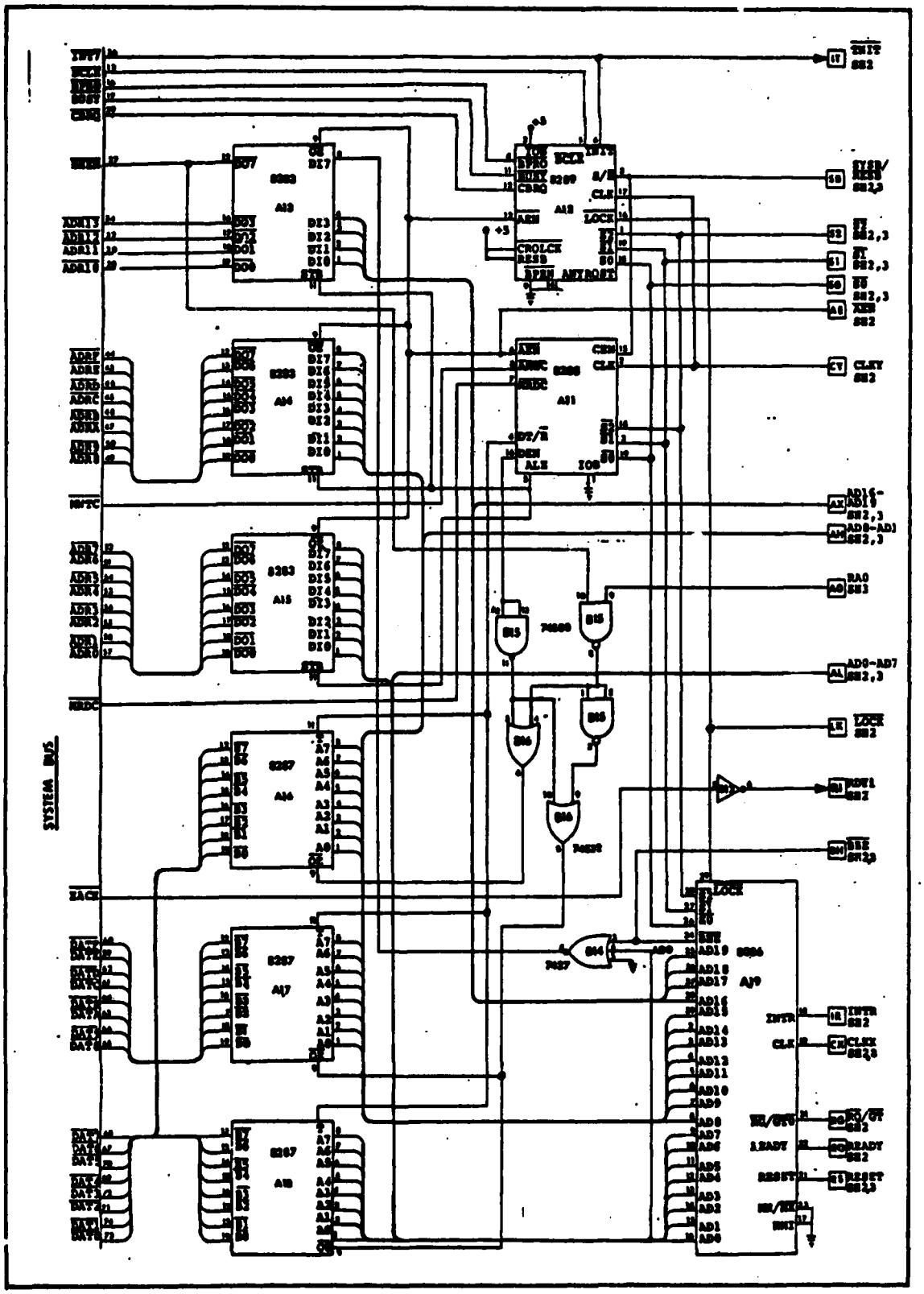


Figure F-1. Schematic Diagram of Demonstration Network Module (Sheet 1 of 3)

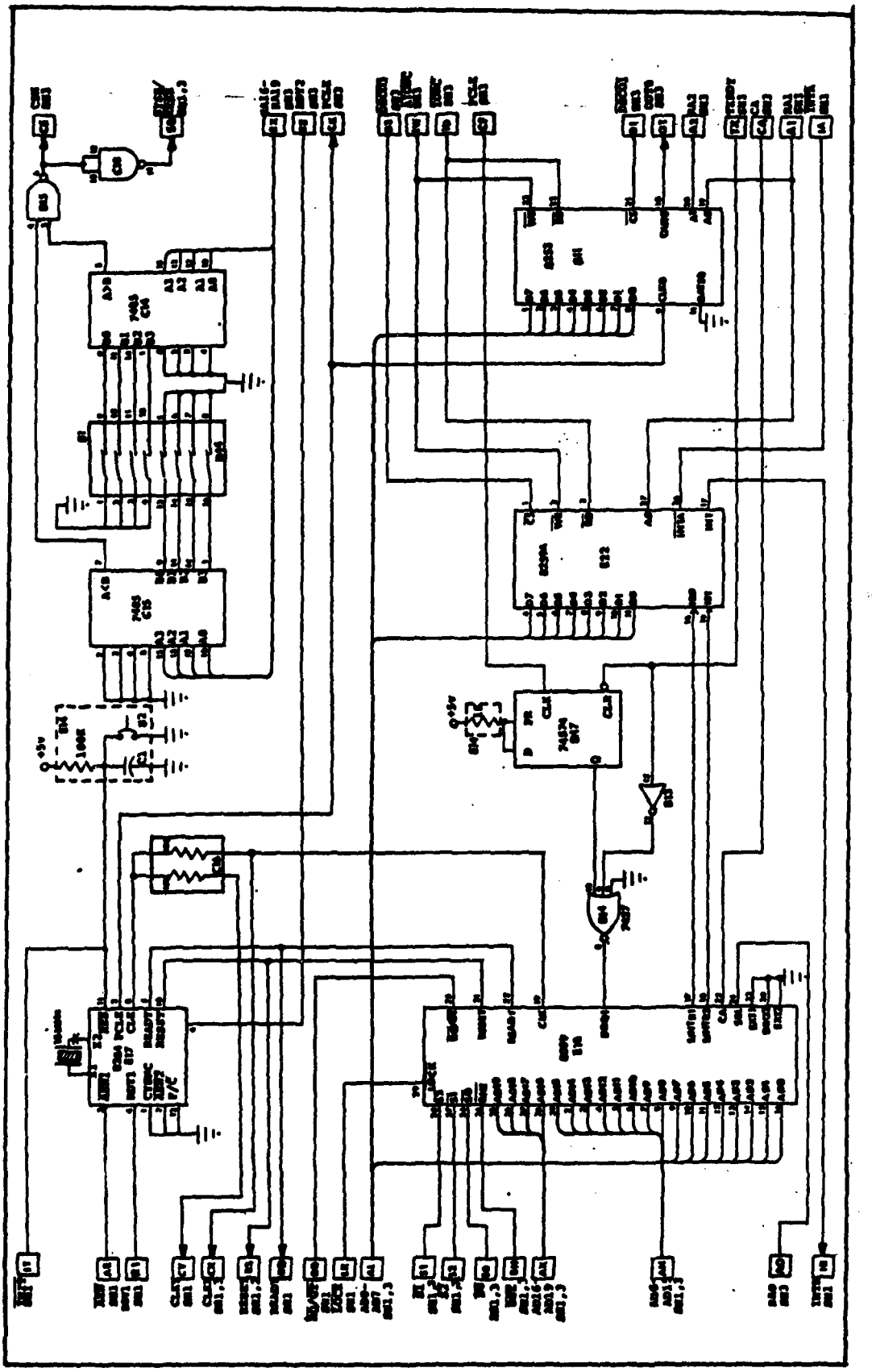


Figure F-2. Schematic Diagram of Demonstration Network Module (Sheet 2 of 3)

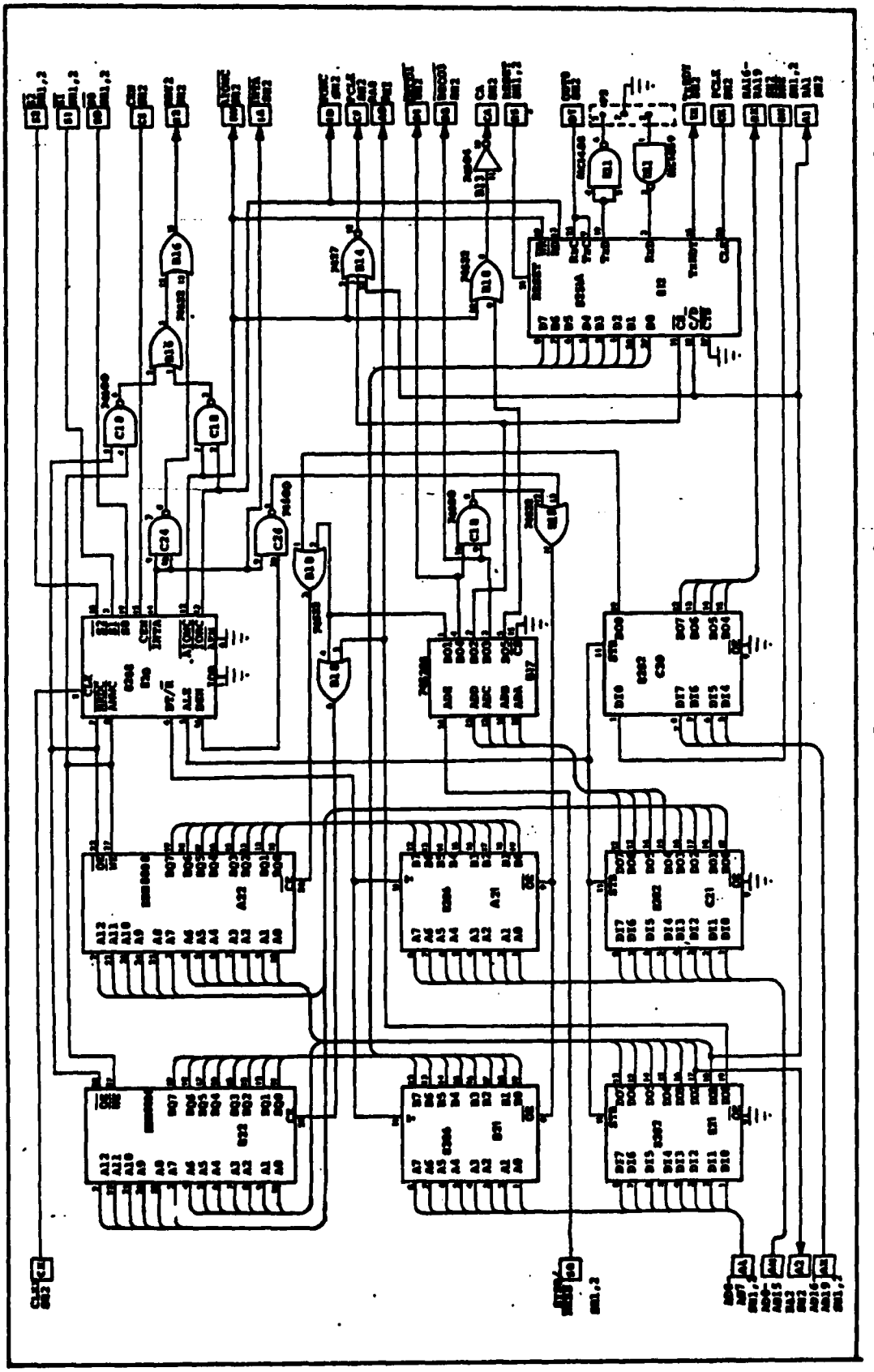


Figure F-3. Schematic Diagram of Demonstration Network Module (Sheet 3 of 3)

Appendix G

Notes on the use of the ICE-86A

The In-Circuit Emulator for the 8086/88 (ICE-86A) proved to be an invaluable aid for testing the UNID II. The operators manual (Ref 23) is written well, but there were some aspects of ICE which are not stressed, or they were difficult to find. Some notes are listed below about the ICE-86A:

- A correction cannot be made to a MACRO without removing it and retyping it. It is best to use short Macros or to write and correct them using the editor.
- The LIST function should be used during all sessions. It keeps a file of everything that appears on the screen, which can be analyzed later.
- A problem was encountered sometimes when telling ICE to emulate a large block of instructions. Sometimes the 8086 code had to be stepped through a step at a time. After the instructions had been emulated once, then the complete block could be executed with no problems.
- A problem occured in which ICE was giving the "control circuit failure" all the time, even when ICE was booted up. I ran the diagnostic program (Ref 23:2-6), and it failed the RAM ripple checks and the RAM R/W test. I corrected this by replacing the RAM chip located at A73 on the Controller board. The faulty chip was a 211A and I replaced it with a 2111AL-4, which is a slower chip, but it works.
- The SBC 86/12A must have the piggyback DIP clips installed when connected to ICE (Ref 23:F-1).
- The Intel notes, which came with the ICE, say that the SBC 86/12A should have a modification package installed to upgrade it to a revision T. This package includes an 8284A and a 8202A. They have been ordered, but the SBC 86/12A appears to be working fine without them.

Appendix H

Demonstration Program Listings

The program listings for the Demonstration program are listed on the following pages. There are three seperate listings. First, is the PL/M-86 program for the local module, DEMOL. Second is the one for the network module, DEMON, and third, the listing for the 8089, TASK1 (INTTB) and TASK2 (PROGTB). The 8089 program was written in ASM-89, but it was not assembled, because an ASM 89 assembler was not available.

Table of Contents

```
ISIS-II PL/M-86 V2.1 COMPILATION OF MODULE DEMOL
OBJECT MODULE PLACED IN :F1:L8612.OBJ
COMPILER INVOKED BY:  PLM86 :F1:L8612.SRC DEBUG SYMBOLS
                      DATE(4 OCT 82) PAGELENGTH(47) &
                      WORKFILES(:F1:,:F1:)


        $TITLE('NETWORK MODULE DEMOL') LARGE OPTIMIZE(2)
1       DEMOL:    DO;
        /*********************************************************/
        /*                                                       */
        /*  PLM86 DEMONSTRATION PROGRAM FOR THE UNID II          */
        /*     NETWORK MODULE PROTOTYPE: LOCAL SETUP             */
        /*                                                       */
        /*********************************************************/

        /*********************************************************/
        /*                                                       */
        /*                LITERAL DECLARATIONS                   */
        /*                                                       */
        /*********************************************************/

        /* LOCAL 8259A */

2  1    DECLARE
          INT$STAT$PORTL   LITERALLY   '0C0H',
          INT$MSK$PORTL    LITERALLY   '0C2H',
          INT$ICW1$L       LITERALLY   '13H',
          INT$ICW2$L       LITERALLY   '50H',
          INT$ICW4$L       LITERALLY   '0FH',
          INT$MASK$L       LITERALLY   '0FFH';


       /* LOCAL 8255 LITERALS */
       /* THIS PPI IS ON THE LOCAL BOARD. IT WILL BE USED*/
       /* TO OUTPUT A PULSE WHICH RESETS THE NETWORK     */
       /* MODULE */


3  1    DECLARE
          PPI$PORT$CL      LITERALLY   '0CCH',
          PPI$CNTRL$PORT   LITERALLY   '0CEH',
          PPI$CL$OUTPUT    LITERALLY   '9AH',
          RESET$NETWORK    LITERALLY   '01H';
```

```
PL/M-86 COMPILER   NETWORK MODULE DEMOL   4 OCT 82


      /*  MISCELLANOUS DECLARATIONS  */

 4  1   DECLARE
           TRUE      LITERALLY    '0FFH',
           FALSE     LITERALLY    '00H';


       /*****************************************************/
       /*     THIS IS THE MAIN-PROGRAM WHICH INITILIZES     */
       /* (BY RESET), THE NETWORK 8086 AND THE 8089. THE    */
       /* 8086/L PERFORMS THE INITILIZATION OF THE LOCAL    */
       /* 8259A AND IT ALSO RESETS THE NETWORK MODULE.      */
       /*****************************************************/


       /************** INITILIZATION ********************/


 5  1   LOCAL$START:    DISABLE;
 6  1           OUTPUT(INT$STAT$PORTL) = INT$ICW1$L;
 7  1           OUTPUT(INT$MSK$PORTL)  = INT$ICW2$L;
 8  1           OUTPUT(INT$MSK$PORTL)  = INT$ICW4$L;
 9  1           OUTPUT(INT$MSK$PORTL)  = INT$MASK$L;


PL/M-86 COMPILER   NETWORK MODULE DEMOL  4 OCT 82


 10 1                 OUTPUT(PPI$CNTRL$PORT) = PPI$CL$OUTPUT;
 11 1                 OUTPUT(PPI$PORT$CL)    = RESET$NETWORK;
 12 1                 CALL TIME(5);
                                  /*GIVES A DELAY OF 500us*/
 13 1                 OUTPUT(PPI$PORT$CL)    = 00H;

14 1          LINIT:  DO WHILE TRUE<>FALSE;
15 2           END;


 16 1   STOP:    END DEMOL;
```

PL/M-86 COMPILER NETWORK MODULE DEMOL 4 OCT 82

SYMBOL LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
1	0004H	68	DEMOL. . . . . PROCEDURE STACK=0000H
4			FALSE. . . . . LITERALLY
2			INTICW1L . . . . . LITERALLY
2			INTICW2L . . . . . LITERALLY
2			INTICW4L . . . . . LITERALLY
2			INTMASKL . . . . . LITERALLY
2			INTMSKPORTL. . . . LITERALLY
2			INTSTATPORTL . . . LITERALLY
14	0040H		LINIT. . . . . LABEL
5	0015H		LOCALSTART . . . . LABEL
3			PPICLOUTPUT. . . . LITERALLY
3			PPICNTRLPORT . . . LITERALLY
3			PPIPORTCL. . . . . LITERALLY
3			RESETNETWORK . . . LITERALLY
16	0046H		STOP . . . . . LABEL
4			TRUE . . . . . LITERALLY

MODULE INFORMATION:

```
CODE AREA SIZE      = 0048H     72D
CONSTANT AREA SIZE  = 0000H      0D
VARIABLE AREA SIZE  = 0000H      0D
MAXIMUM STACK SIZE  = 0000H      0D
91 LINES READ
0 PROGRAM ERROR(S)
```

END OF PL/M-86 COMPILATION

```
PL/M-86 COMPILER     NETWORK MODULE DEMON

ISIS-II PL/M-86 V2.1 COMPILATION OF MODULE DEMON
OBJECT MODULE PLACED IN N8612.OBJ
COMPILER INVOKED BY:  :F1:PLM86 N8612.SRC DEBUG SYMBOLS
WORKFILES(:F0:,:F0:) DATE(4 OCT 82)



$TITLE('NETWORK MODULE DEMON') LARGE OPTIMIZE(2)
DEMON:   DO;
/*************************************************************/
/*                                                           */
/*     PLM86 DEMONSTRATION PROGRAM FOR THE UNID II           */
/*       NETWORK MODULE PROTOTYPE: NETWORK MOD.              */
/*                                                           */
/*************************************************************/

/*************************************************************/
/*                                                           */
/*                LITERAL DECLARATIONS                       */
/*                                                           */
/*************************************************************/


/* RAM LOCATIONS FOR THE NETWORK 8086 */

   DECLARE
    SINT$BASE    LITERALLY   'OFFFF6H', /*SYS INITILIZATION BLOCK */
    SCB$BASE     LITERALLY   'OFFFE0H', /*SYSTEM CONTROL BLOCK */
    CB$BASE      LITERALLY   'OFFFD0H', /* COMMAND BLOCK */
    PB$BASE      LITERALLY   'OFF000H', /* PARAMETER BLOCK */
    TB$BASE      LITERALLY   'OFF100H', /* TASK BLOCK */
    MSG$BASE     LITERALLY   'OFF300H', /* DISPLAY MESSAGE BUFFER */
    INTR$TYPE    LITERALLY   '0140H'  ; /* INTERRUPT VECTOR TABLE */

/* NETWORK 8259A LITERALS */

   DECLARE
    INT$STAT$PORT   LITERALLY   '06000H',
    INT$MASK$PORT   LITERALLY   '06002H',
    INT$ICW1        LITERALLY   '13H',   /* SINGLE PIC USED */
                                         /* ICW4 NEEDED     */
                                         /* EDGE TRIGERRED  */
    INT$ICW2        LITERALLY   '50H',   /* VECTORING BYTE  */
    INT$ICW4        LITERALLY   '07H',   /* 8086/88 MODE    */
                                         /* AUTOMATIC EOI   */
                                         /* NONBUFFERED MODE*/
    INT$MASK        LITERALLY   'OFEH';  /* IRO UNMASKED    */


/* RAM LOCATIONS FOR THE 8089 */

   DECLARE
    SCB$89   LITERALLY   'OFFFE0H', /* SYSTEM CONTROL BLOCK */
    CB$89    LITERALLY   'OFFFD0H', /* COMMAND BLOCK */
    PB$89    LITERALLY   'OFF000H', /* PARAMETER BLOCK */
    TB$89    LITERALLY   'OFF100H', /* TASK BLOCK */
    MSG$89   LITERALLY   'OFF300H'; /* DISPLAY MESSAGE BUFFER */
```

```
/* 8089 CCW'S */

   DECLARE
    INIT$CCW     LITERALLY   '13H', /* I/O INITILIZATION CCW */
                                    /* ENABLE INTERRUPTS     */
                                    /* EXECUTE TASK BLOCK IN */
                                    /* SYSTEM MEMORY         */
    DSP$CCW      LITERALLY   '08H'; /* DISPLAY MESSAGE CCW   */
                                    /* RESET INTERRUPT     */
                                    /* EXECUTE TASK BLOCK  */
                                    /* IN SYSTEM MEMORY    */


/* 8089 INITILIZATION COMMANDS */

   DECLARE
    SOC$CMD      LITERALLY   '00H', /* 8 BIT I/O BUS */
    SYSBUS$CMD   LITERALLY   '01H'; /* 16 BIT SYSTEM BUS */


/* 8089 CHANNEL ATTENTION */

   DECLARE
    CHAN1$ATT    LITERALLY   '7000H', /* CA=1, SEL=0 */
    CHAN2$ATT    LITERALLY   '7001H'; /* CA=1, SEL=1 */


/* MISCELLANEOUS DECLARATIONS */

   DECLARE
    BUSYSTATUS       LITERALLY   'OFFH',
    CR               LITERALLY   'ODH',
    E                LITERALLY   '45H',
    EOT              LITERALLY   '04H',
    ESC              LITERALLY   '1BH',
    FALSE            LITERALLY   '00H',
    LF               LITERALLY   'OAH',
    NMBR$MASK        LITERALLY   '07H',
    TRUE             LITERALLY   'OFFH';
```

```
/*****************************************************************************/
/*                                                                           */
/*                              RAM DECLARATIONS                             */
/*                                                                           */
/*****************************************************************************/


DECLARE SINT STRUCTURE (SYSBUS WORD, SCB$PTR POINTER) AT (SINT$BASE);

        /*************************************************/
        /*                          \     SYSBUS COMMAND   */
        /*************************************************/
        /*          SCB             OFFSET               */
        /*************************************************/
        /*          SCB             SEGMENT              */
        /*************************************************/


DECLARE SCB STRUCTURE (SOC WORD, CB$PTR POINTER) AT (SCB$BASE);

        /*************************************************/
        /*                          \     SOC COMMAND      */
        /*************************************************/
        /*       COMMAND  BLOCK  OFFSET                  */
        /*************************************************/
        /*       COMMAND  BLOCK  SEGMENT                 */
        /*************************************************/


DECLARE CB(2) STRUCTURE (CCW BYTE, BUSY BYTE, PB$PTR POINTER,
          DUMMY WORD) AT (CB$BASE);

        /*************************************************/
        /*      BUSY  FLAG          \          CCW       */
        /*************************************************/
        /*          PARAMETER  BLOCK  OFFSET             */
        /*************************************************/
        /*          PARAMETER  BLOCK  SEGMENT            */
        /*************************************************/
        /*                DUMMY  WORD                    */
        /*************************************************/
/* THE ABOVE CB ARRAY CONTAINS TWO STRUCTURES; */
/* ONE FOR EACH CHANNEL OF THE 8089 */

DECLARE     PB  STRUCTURE (TB$PTR POINTER, MSG$PTR POINTER,
                  LEVEL BYTE, CI BYTE) AT (PB$BASE);

        /*************************************************/
        /*           TASK  BLOCK  OFFSET                 */
        /*************************************************/
        /*           TASK  BLOCK  SEGMENT                */
        /*************************************************/
        /*           MESSAGE  BUFFER  OFFSET             */
        /*************************************************/
        /*           MESSAGE  BUFFER  SEGMENT            */
        /*************************************************/
        /* CHARACTER FROM CRT \ DISPLAY LEVEL CMD TO IOP */
        /*************************************************/
```

```
DECLARE     TB  (512) BYTE AT (TB$BASE);

        /*************************************************/
        /*     RAM BUFFER FOR TASK BLOCK PROGRAM         */
        /*************************************************/



DECLARE     MSG$BUF (1024) BYTE AT (MSG$BASE);

        /*************************************************/
        /*       DISPLAY  MESSAGE BUFFER                 */
        /*************************************************/


DECLARE     INTR$VEC$80 POINTER AT (INTR$TYPE);

DECLARE     INTR$IP$80  WORD    AT (INTR$TYPE);




/*****************************************************************************/
/*                                                                           */
/*                  ROM DECLARATION AND INITILIZATION                        */
/*                                                                           */
/*****************************************************************************/

   DECLARE  MENU(*) BYTE DATA (CR,LF,ESC,E,
'                     ************************************',CR,LF,
'                     *                                  *',CR,LF,
'                     *  THE 8086 AND A NETWORK MODULE   *',CR,LF,
'                     *          DEMONSTRATION           *',CR,LF,
'                     *                                  *',CR,LF,
'                     ************************************',CR,LF,
CR,LF,LF,
'               SELECTION          TOPIC',CR,LF,LF,
'                  1              WHAT IS THE 8087 IOP ?',CR,LF,LF,
'                  2              WHAT IS THE 8287 BUS ARBITRATOR ?',CR,LF,LF,
'                  3              ABOUT THIS DEMONSTRATION',CR,LF,LF,B
'                  5              8089 COMMUNICATION PROTOCOL ',CR,LF,LF,
LF,LF,
'       FOR ADDITIONAL INFORMATION ON THE ABOVE TOPICS',CR,LF,
'       PLEASE SELECT THE APPROPRIATE ENTRY (1,2,3,4,5) -',EOT,EOT);
```

```
DECLARE MSG1(*) BYTE DATA(CR,LF,ESC,E,
'                              8089 I/O PROCESSOR',
CR,LF,LF,LF,
'     THE 8089 I/O PROCESSOR IS A FIRST OF ITS KIND SYSTEMS COMPONENT.  IT',
CR,LF,LF,
'USES THE CONCEPT OF A CHANNEL CONTROLLER, COMMON IN MAINFRAMES, TO SOLVE',
CR,LF,LF,
'THE I/O PROCESSING AND HIGH PERFORMANCE DMA REQUIREMENTS OF MICROPROCESSOR',
CR,LF,LF,
'SYSTEMS.  THE 8089 CAN BE USED IN CONJUNCTION WITH THE 8086 (16 BIT BUS)',
CR,LF,LF,
'OR 8088 (8 BIT BUS) AND 8 OR 10 PERIPHERALS TO SIGNIFICANTLY ENHANCE',
CR,LF,LF,
'SYSTEM PERFORMANCE.  THE 8089 OFF LOADS I/O FROM THE HOST CPU AND PROCESSES',
CR,LF,LF,
'CONCURRENTLY WITH CPU ACTIVITY.  ALSO, THE 8087 ADDS INTELLIGENCE TO THE',
CR,LF,LF,
'PERPHERIAL SUBSYSTEM WHILE MODULARIZING AND SIMPLIFYING THE SYSTEM I/O.',
CR,LF,LF,
'EACH IOP HAS TWO I/O CHANNELS THAT CAN PROVIDE DMA AT 1.25 MEGABYTE/SEC.',
CR,LF,LF,
'PROCESS INDEPENDENT PROGRAMS, AND HANDLE MULTIPLE I/O DEVICES.',
CR,LF,LF,
'                    TO SELECT ANOTHER MESSAGE TYPE Y-',EOT,EOT);


DECLARE MSG2(*) BYTE DATA (CR,LF,ESC,E,
'                            THE 8289 BUS ARBITER',
CR,LF,LF,LF,
'      THE 8289 BUS ARBITER PROVIDES THE HARDWARE MECHANISMS FOR INTER-',
CR,LF,LF,
'PROCESSOR COMMUNICATION AND SHARED RESOURCES IN A MULTIPLE CPU SYSTEM.  THE',
CR,LF,LF,
'8289 FEATURES SEVERAL USER DEFINABLE PRIORITIZATION AND BUS CONFIGURATIONS.',
CR,LF,LF,
'DEMONSTRATED HERE, THE RESB MODE SEPERATES 86/12 RESOURCES FROM',
CR,LF,LF,
'SYSTEM BUS SHARED RESOURCES.  THE RESIDENT BUS ALLOWS BOTH MEMORY AND I/O',
CR,LF,LF,
'ACCESSES.  THE 8289 COMPLETELY ARBITRATES SYSTEM BUS USAGE TO MANAGE',
CR,LF,LF,
'MULTIPLE PROCESSOR CONTENTION.',
CR,LF,LF,
'       THE 8086 FAMILY AND MULTIBUS CONCEPT ALLOWS PARTITIONING APPLICATIONS',
CR,LF,LF,
'INTO SMALLER MORE MANAGEABLE TASKS.  THUS, ADDING NEW FUNCTIONS OR UPGRADING',
CR,LF,LF,
'EXISTING ONES WILL HAVE MINIMAL EFFECT ON THE ORIGINAL DESIGN.',
CR,LF,LF,
'                    TO SELECT ANOTHER MESSAGE TYPE Y-',EOT);
```

```
DECLARE MSG3(*) BYTE DATA(CR,LF,ESC,E,
'                        ABOUT THIS DEMONSTRATION',
CR,LF,LF,
'       IN THIS DEMONSTRATION AN SBC 86/12A AND THE NETWORK MODULE',
CR,LF,LF,
'ARE INTERFACED VIA THE INTEL MULTIBUS.  THE 86/12A SERVES AS THE LOCAL',
CR,LF,LF,
'MODULE.  IN THIS DEMO THE 8089 UNBURDENS THE 8086 BY HANDLING MESSAGE',
CR,LF,LF,
'TRANSFERS TO THE CRT AND PROCESSING MESSAGE REQUESTS.  OPERATION IS AS',
CR,LF,LF,
'FOLLOWS:   USING A CHANNEL ATTENTION (CA) THE 8086 INITILIZES THE 8089 AND',
CR,LF,LF,
'CAUSES IT TO EXECUTE A TASK BLOCK TO PROGRAM THE PERIPHERAL DEVICES',
CR,LF,LF,
'ON ITS RESIDENT BUS.  THE 8089 THEN INTERRUPTS THE 8086 TO REQUEST A MESSAGE',
CR,LF,LF,
'FOR DISPLAY.  RESPONDING, THE 8086 SETS UP LINKAGE TO THE TASK BLOCK',
CR,LF,LF,
'PROGRAM AND ISSUSES A CA TO THE 8089.  AFTER EACH CA THE 8089 DISPLAYS THE',
CR,LF,LF,
'MESSAGE, POLLS THE CRT TERMINAL FOR A VALID MESSAGE REQUEST AND THEN',
CR,LF,LF,
'INTERRUPTS THE 8086  THE CYCLE IS REPEATED, ',
CR,LF,LF,
'                    TO SELECT ANOTHER MESSAGE TYPE Y-',EOT,EOT);


/*  8089 PROGRAM - TASK1 */

   DECLARE  INITTB(128) BYTE DATA(31H,30H,2,50H,8,4DH,0CAH,0,0,0,0,
   8,4DH,40H,0,0,0,0,8,4DH,0CAH,0,0,0,0,8,4DH,25H,31H,30H,6,40H,8,4DH,
   37H,31H,30H,0,40H,8,4DH,16H,8,4DH,0,0AH,4FH,9,59H,40H,0,20H,48H);


/*  8089 PROGRAM - TASK2  */

   DECLARE  PROCTB(128) BYTE DATA(0D1H,30H,2,50H,3,8BH,4,31H,30H,0,50H,
   0F1H,30H,4,0FFH,0C0H,0,60H,0,51H,30H,2,50H,0AH,0E7H,8,14H,0F1H,30H,59H,
   0FFH,28H,0BAH,0FDH,8,0B5H,0FAH,0AH,4FH,9,59H,88H,20H,25H,0F1H,30H,37H,
   0F0H,26H,0DAH,0FDH,0,91H,2,0CFH,9,0AH,0B7H,9,0F4H,0F1H,30H,37H,0FCH,0AH,
   0B3H,9,0E8H,0F1H,30H,30H,0FFH,0AH,0B3H,9,0,0CDH,40H,0,20H,48H);
```

```
DECLARE MSG4(*) BYTE  DATA (CR,LF,ESC,E,
'                    8089 INITILIZATION PROTOCOL',
CR,LF,LF,LF,
'SYSTEM INITILIZATION       ************************************************',
CR,LF,
'                           *                        *       SYSTEM BUS      *',
CR,LF,
'                           ************************************************',
CR,LF,
'                           *        SYSTEM CONTROL BLOCK ADDRESS            *',
CR,LF,
'                           ************************************************',
CR,LF,
'SYSTEM CONTROL BLOCK       ************************************************',
CR,LF,
'                           *                        *       SOC COMMAND     *',
CR,LF,
'                           ************************************************',
CR,LF,
'                           *       COMMAND  BLOCK  ADDRESS                  *',
CR,LF,
'                           ************************************************',
CR,LF,LF,LF,
'       ON THE FIRST CHANNEL ATTENTION AFTER RESET, THE IOP READS THESE',
CR,LF,LF,
'CONTROL BLOCKS TO DETERMINE THE WIDTH OF THE SYSTEM BUS (8 OR 16), THE',
CR,LF,LF,
'I/O BUS WIDTH (8 OR 16), PRIORITY INFORMATION, AND WHERE TO FIND INFORMATION',
CR,LF,LF,
'DEFINING SUBSEQUENT CHANNEL ATTENTIONS (THE COMMAND CONTROL BLOCK). ',
CR,LF,LF,
'                    TO SELECT ANOTHER MESSAGE TYPE Y-',EOT,EOT);
```

```
DECLARE     MSGS(*) BYTE   DATA (CR,LF,ESC,E,
'                           8089 TASK COMMUNICATION PROTOCOL',
CR,LF,LF,LF,
'                           ************************************************',
CR,LF,
'COMMAND BLOCK              *     BUSY FLAG          *  CHANNEL COMMAND WORD *',
CR,LF,
'                           ************************************************',
CR,LF,
'    (ONE PER CHANNEL)      *           PARAMETER BLOCK ADDRESS              *',
CR,LF,
'                           ************************************************',
CR,LF,LF,
'  PARAMETER BLOCK          ************************************************',
CR,LF,
'                           *             TASK  BLOCK  ADDRESS               *',
CR,LF,
'                           ************************************************',
CR,LF,
'                           *           USER DEFINED MESSAGE AREA            *',
CR,LF,
'                           ************************************************',
CR,LF,LF,
'   TASK BLOCK              ************************************************',
CR,LF,
'                           *    TASK PROGRAM TO BE EXECUTED BY THE 8089     *',
CR,LF,
'                           ************************************************',
CR,LF,LF,
'     AFTER A CHANNEL ATTENTION, THE 8089 READS THESE BLOCKS TO SEE WHAT THE',
CR,LF,LF,
'CPU WANTS (CHANNEL ATTENTION WORD) AND WHERE TO FIND ADDITIONAL INFORMATION',
CR,LF,LF,
'(PARAMETER BLOCK). THE PARAMETER BLOCK GIVES THE TASK PROGRAM ADDRESS AND',
CR,LF,LF,
'PARAMETERS TO BE PASSED.    TO SELECT ANOTHER MESSAGE TYPE Y-',EOT,EOT);
```

```
/*************************************************************/
/*     THIS IS THE MAIN PROGRAM IN WHICH THE 8086N PER-      */
/* FORMS THE INITILIZATION OF THE 8259A AND IT INITIL-       */
/* IZES THE VECTOR INTERRUPT TABLE.  THE 8086/N THEN         */
/* ISSUES A CHANNEL ATTENTION TO THE 8039 WHICH INITIL-      */
/* IZES THE 8251A AND 8253. AFTER ALL INITILIZATIONS ARE     */
/* COMPLETE, THE PROGRAM IS TOTALLY INTERRUPT DRIVEN FROM    */
/* THE 8089.  THE 8089 INTERRUPTS THE 8086/N TO REQUEST A    */
/* NEW MESSAGE FOR DISPLAY.  TO SERVICE THE INTERRUPT,       */
/* THE 8086/N TRANSFERS THE NEW MESSAGE INTO THE MESSAGE     */
/* BUFFER, SETS UP THE APPROPRIATE TASK BLOCK PROGRAM        */
/* AND ISSUES A NEW CA TO THE IOP TO ALLOW IT TO DIS-        */
/* PLAY THE NEW MESSAGE.  THE 8086/N WILL HALT AFTER         */
/* ISSUING THE CA AND WAIT FOR THE NEXT MESSAGE REQUEST      */
/*                                                           */
/*    AFTER EACH CA, THE 8089 WILL DISPLAY THE REQUESTED     */
/* MESSAGE. THEN POLL FOR A NEXT MESSAGE REQUEST ENTERED     */
/* AT THE CRT.  UPON RECEIVING A VALID REQUEST, THE 8089     */
/* RETURNS THE REQUEST TO THE 8086/N, ISSUES AN INTR-        */
/* RUPT TO THE 8086/N AND HALTS ITS CURRENT TB EXECUTION.    */
/* THE 8089 PERFORMS NO OTHER ACTIVITIES UNTIL AWAKENED      */
/* BY THE CA FROM THE 8086/N TO DISPLAY THE NEXT MESSAGE.    */
/*************************************************************/



/************MESSAGE PROCESSING INTERRUPT ROUTINE*********/

MSGDSPL:  PROCEDURE INTERRUPT 80 PUBLIC;

          IF PB.CI='Y' THEN              /* 'Y' INDICATES MENU IS REQUESTED */
              DO;
                CALL MOVB(@MENU, @MSG$BUF,SIZE(MENU)); /* MOVE MENU INTO */
                                                       /* MESSAGE BUFFER */
                PB.LEVEL = FALSE;
              END;
            ELSE DO;
                PB.LEVEL = TRUE;
                DO CASE (PB.CI AND NMBR$MASK)-1;    /* DETERMINE WHICH MESSAGE */
                                                    /* WAS REQUESTED           */
                  CALL MOVB (@MSG1, @MSG$BUF, SIZE (MSG1));
                  CALL MOVB (@MSG2, @MSG$BUF, SIZE (MSG2));
                  CALL MOVB (@MSG3, @MSG$BUF, SIZE (MSG3));
                  CALL MOVB (@MSG4, @MSG$BUF, SIZE (MSG4));
                  CALL MOVB (@MSG5, @MSG$BUF, SIZE (MSG5));
CEND:           END;
              END;


          CALL MOVB(@PROGTB, @TB, SIZE(PROGTB)); /* MOVE THE 8089 TASK2 */
                                                 /* INTO THE TASK BLOCK */
          PB.TB$PTR = TB$89;                     /* SET THE POINTER TO  */
                                                 /* THE TASK BLOCK      */
          PB.MSG$PTR = MSG$89;                   /* SET THE POINTER TO  */
                                                 /* THE MESSAGE BUFFER  */

          CB(0).CCW = DSP$CCW;
          CB(0).PB$PTR = PB$89;                  /* SET THE POINTER TO  */
                                                 /* PARAMETER BLOCK     */

          RETURN;

MSGEND:   END MSGDSPL;
```

```
/***************** INITILIZATION ******************************/


 NETWORK$START: DISABLE;

    INTR$VEC$80 = @MSGDSPL;  /* INITILIZE INTERRUPT VECTOR TABLE */
    INTR$IP$80 = INTR$IP$80 - 27;

L8259:  OUTPUT(INT$STAT$PORT) = INT$ICW1; /* INITILIZE THE PIC */
        OUTPUT(INT$MASK$PORT) = INT$ICW2;
        OUTPUT(INT$MASK$PORT) = INT$ICW4;
        OUTPUT(INT$MASK$PORT) = INT$MASK;

INIT89: SINT.SYSBUS = SYSBUS$CMD; /* SET PHYSICAL SYS BUS WIDTH = 16 */
        SINT.SCB$PTR =.SCB$89;
XSCB:   SCB.SOC = SOC$CMD;        /* SET PHYSICAL I/O BUS WIDTH = 8 */
                                  /* 8089 RQ/GT = MODE 0 */
        SCB.CB$PTR = CB$89;       /* SET POINTER TO COMMAND BLOCK */

        CB(0).BUSY = BUSYSTATUS;  /* SET BUSY FLAG */
OT891:  OUTPUT(CHAN2$ATT) = 0;    /* INITILIZE THE 8089 */
                                  /* THE 8089 WILL NOW PERFORM ITS INTERNAL */
                                  /* ROM INITILIZATION ROUTINE.  WHEN IT IS */
                                  /* FINISHED IT WILL RESET THE BUSY FLAG   */

NBLP:   DO WHILE CB(0).BUSY = BUSYSTATUS; /* WAIT FOR BUSY FLAG TO BE RESET */
        END;

        CALL MOVB(@INITTB,@TB,SIZE(INITTB)); /* MOVE TASK1 INTO */
                                             /* TASK BLOCK      */

        CB(0).CCW = INIT$CCW;     /* SET 8089 CHANNEL COMMAND WORD */
                                  /* ENABLE INTERRUPTS */
                                  /* START CHANNEL PROGRAM LOCATED */
                                  /*    IN SYSTEM SPACE            */
        CB(0).PB$PTR = PB$89;     /* SET POINTER TO PARAMETER BLOCK */
        PB.TB$PTR = TB$89;        /* SET POINTER TO TASK BLOCK */

OT892:  OUTPUT(CHAN1$ATT) = 0;    /* ISSUE SECOND CA TO 8089 */
                                  /* 8089 WILL EXECUTE TASK1 */


        INTR$LOOP:  HALT;         /* WAIT FOR INTERRUPT */

OT893:  OUTPUT(CHAN1$ATT) = 0;    /* ISSUE CA TO 8089 */
                                  /* 8089 WILL NOW EXECUTE TASK2 */
        GOTO INTR$LOOP;


STOP:   END DEMON;
```

SYMBOL LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
11	0001H	1	BUSY . . . . . BYTE MEMBER(CB)
8			BUSYSTATUS . . . . LITERALLY
11	FHFFD0H	16	CB . . . . . STRUCTURE ARRAY(2) AT ABSOLUTE
4			CB09 . . . . . LITERALLY
2			CBBASE . . . . . LITERALLY
10	0002H	4	CBPTR. . . . . POINTER MEMBER(SCB)
11	0000H	1	CCW. . . . . BYTE MEMBER(CB)
39	19D7H		CEND . . . . . LABEL
7			CHAN1ATT . . . . . LITERALLY
7			CHAN2ATT . . . . . LITERALLY
12	0009H	1	CI . . . . . BYTE MEMBER(PB)
8			CR . . . . . LITERALLY
1	1820H	250	DEMON. . . . . PROCEDURE STACK=0024H
5			DSPCCW . . . . . LITERALLY
11	0006H	2	DUMMY. . . . . WORD MEMBER(CB)
8			E. . . . . LITERALLY
8			EOT. . . . . LITERALLY
8			ESC. . . . . LITERALLY
8			FALSE. . . . . LITERALLY
55	185DH		INIT89 . . . . . LABEL
5			INITCCW. . . . . LITERALLY
23	16FCH	128	INITTB . . . . . BYTE ARRAY(120) DATA
3			INT1CW1. . . . . LITERALLY
3			INT1CW2. . . . . LITERALLY
3			INT1CW4. . . . . LITERALLY
3			INTMASK. . . . . LITERALLY
3			INTMASKPORT. . . . LITERALLY
16	0140H	2	INTRIP80 . . . . . WORD AT ABSOLUTE
68	1916H		INTRLOOP . . . . . LABEL
2			INTRTYPE . . . . . LITERALLY
15	0140H	4	INTRVEC80. . . . . POINTER AT ABSOLUTE
3			INTSTATPORT. . . . LITERALLY
51	184BH		L8259. . . . . LABEL
12	0008H	1	LEVEL. . . . . BYTE MEMBER(PB)
8			LF. . . . . LITERALLY
17	0000H	794	MENU . . . . . BYTE ARRAY(794) DATA
18	031AH	848	MSG1 . . . . . BYTE ARRAY(848) DATA
19	064AH	802	MSG2 . . . . . BYTE ARRAY(802) DATA
20	098CH	881	MSG3 . . . . . BYTE ARRAY(881) DATA
21	0CFDH	1181	MSG4 . . . . . BYTE ARRAY(1181) DATA
22	119AH	1378	MSG5 . . . . . BYTE ARRAY(1378) DATA
4			MSG89. . . . . LITERALLY
2			MSGBASE. . . . . LITERALLY
14	FHF300H	1024	MSGBUF . . . . . BYTE ARRAY(1024) AT ABSOLUTE
25	1935H	306	MSGDSPL. . . . . PROCEDURE PUBLIC INTERRUPT(80) STACK=0008H
47	1A4CH		MSGEND . . . . . LABEL
12	0004H	4	MSGPTR . . . . . POINTER MEMBER(PB)
61	1865H		NBLP . . . . . LABEL
48	1831H		NETWORKSTART . . . LABEL

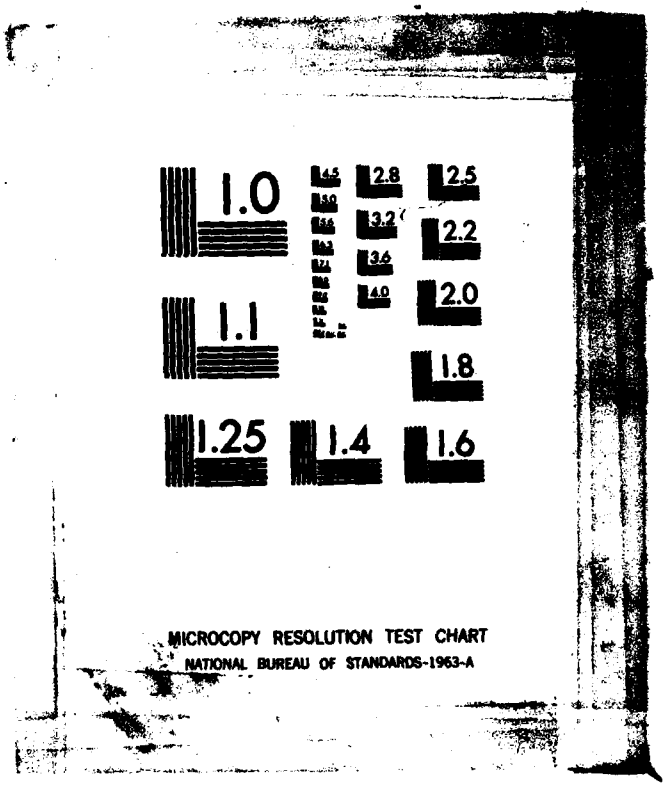
```
  8                    NMBRMASK . . . . .    LITERALLY
 60   18AFH           DT891. . . . . .      LABEL
 67   1910H           DT892. . . . . .      LABEL
 69   1918H           DT893. . . . . .      LABEL
 12 FH1F000H    10    PB . . . . . .        STRUCTURE AT ABSOLUTE
  4                    PB89 . . . . . .      LITERALLY
  2                    PBBASE . . . . . .    LITERALLY
 11   0002H      4    PBPTR. . . . . .      POINTER MEMBER(CB)
 24   177CH    128    PROGTB . . . . . .    BYTE ARRAY(120) DATA
 10 FHFFE0H     6    SCB. . . . . .        STRUCTURE AT ABSOLUTE
  4                    SCB89. . . . . .      LITERALLY
  2                    SCBBASE. . . . . .    LITERALLY
  9   0002H      4    SCBPTR . . . . . .    POINTER MEMBER(SINT)
  9 FHFFF6H     6    SINT . . . . . .      STRUCTURE AT ABSOLUTE
  2                    SINTBASE . . . . .    LITERALLY
 10   0000H      2    SOC. . . . . .        WORD MEMBER(SCB)
  6                    SOCCMD . . . . . .    LITERALLY
 71   191AH           STOP . . . . . .      LABEL
  9   0000H      2    SYSBUS . . . . . .    WORD MEMBER(SINT)
  6                    SYSBUSCMD. . . . .    LITERALLY
 13 FHF100H   512    TB . . . . . .        BYTE ARRAY(512) AT ABSOLUTE
  4                    TB89 . . . . . .      LITERALLY
  2                    TBBASE . . . . . .    LITERALLY
 12   0000H      4    TBPTR. . . . . .      POINTER MEMBER(PB)
  8                    TRUE . . . . . .      LITERALLY
 57   1881H           XSCB . . . . . .      LABEL


MODULE INFORMATION:

     CODE AREA SIZE     = 1A4CH   67320
     CONSTANT AREA SIZE = 0000H       0D
     VARIABLE AREA SIZE = 0000H       0D
     MAXIMUM STACK SIZE = 0024H      36D
     471 LINES READ
     0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION
```

```
;*************************************************************
;*                                                           *
;*        ASM89 DEMO PROGRAM FOR THE NETWORK MODULE          *
;*                                                           *
;*************************************************************
;
NAME   DEMO89
DEMO   SEGMENT
;
;
PUBLIC          INITTB
PUBLIC          PROGTB
;
;
;*************************EQUATES*****************************
;
DADDRESS_8251 EQU       2000H
CADDRESS_8251 EQU       2001H
MODE_8251       EQU     0CAH
RST_8251        EQU     40H
COMMAND                 EQU     25H
MADDRESS_8253 EQU       3003H
COMODE_8253     EQU     37H
COADDRESS_8253          EQU     3000H
CO_LSB_8253     EQU     16H
CO_MSB_8253     EQU     0
Y               EQU     59H
CI              EQU     9H
CHAN_CONTROL    EQU     5001H
MSG_POINTER     EQU     4H
EOT_COMPARE     EQU     0FF04H
LEV             EQU     8H
Y_COMPARE       EQU     0FF59H
MSG_COMPARE     EQU     0F837H
SIX_SEVEN_COMPARE       EQU     0FE37H
ZERO_COMPARE    EQU     oFF30H
;
;
;***************** TASK1 - INITILIZATION ****************
;
;
INITTB:         MOVI    GB,CADDRESS_8251    ;INITILIZE 8251A
        MOVBI   [GB], MODE_8251             ;COMMAND ADDRESS
        NOP
        NOP
        MOVBI   [GB], RST_8251              ;SOFTWARE RESET
        NOP
        NOP
        MOVBI   [GB], MODE_8251             ;2 STOP,CHAR LEN 7,
        NOP                                 ;X16
        NOP
        MOVBI   [GB], COMMAND_8251          ;REC AND TRANSMIT
INIT53:         MOVI    GB,MADDRESS_8253    ;INITILIZE 8253,
```

AD-A124 834 DESIGN OF A PROTOTYPE UNIVERSAL NETWORK INTERFACE DEVICE USING INTEL 8086.. (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. D E PALMER

3/3

UNCLASSIFIED DEC 82 AFIT/GE/EE/82D-52 F/G 9/2 NL


1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
        MOVBI   [GB],COMODE_8253        ;CNT 0, MODE 3, BCD
        MOVI    GB, C0ADDRESS_8253      ;COUNTER 0 ADDRESS
        MOVBI   [GB],C0_LSB_8253        ;LSB=16   8251A BAUD
        MOVBI   [GB],C0_MSB_8253        ;MSB = 0    RATE GEN
        MOVBI   [PP].CI, Y              ;Y TO CI BYTE IN
                                        ;   PARAMETER BLOCK
                                ;TO SELECT MENU FOR DISPLAY
        SINTR                           ;INTERRUPT 8086/N
        HALT                            ;WAIT FOR CA
;
;
;***************TASK2 - MESSAGE PROCESSING ***************
;
;
PROGTB:         MOVI    CC,CHAN_CONTROL ;LOAD CHAN CNTRL WRD
                                        ;MEMORY TO PORT
                                        ;SYNC ON DESTINATION
                                        ;GA SOURCE
                                        ;TERM ON MASK COMP
                                        ;TERMINATE OFFSET =0
        LPD     GA,[PP].MSG_POINTER     ;MESSAGE POINTER,
                                        ;DMA SOURCE
        MOVI    GB,DADDRESS_8251        ;8251A DATA ADDR,
                                        ;DMA DESTINATION
        MOVI    MC,EOT_COMPARE          ;MASK COMP FOR EOT
        WID     16,8                    ;SOURCE BUS 16,
                                        ;DESTINATION BUS 8
DMA:    XFER                            ;START DMA VIA DRQ1
                                        ;AND 8251A TXRDY
        MOVI    GC,CADDRESS_8251        ;8251A COMMAND AND
                                        ;STATUS ADDRESS

LEVEL:          JZB     [PP].LEV, MSGSEL;CHECK LEVEL BYTE
MENSEL:         MOVI    MC, Y_COMPARE   ;MASK COMPARE FOR Y
RXRDY1:         JNBT    [GC],1,RXRDY1   ;RECEIVE READY ?
      JMCNE     [GB],RXRDY1             ;Y ?
      MOVBI     [PP].CI, Y              ;Y TO CI BYTE
      MOVBI     [GB], Y                 ;ECHO
      JMP       INTR86                  ;
MSGSEL:  MOVI MC,MSG_COMPARE  ;MASK COMP FOR MESSAGE SEL
RXRDY2:         JNBT    [GC],1, RXRDY2  ;RECEIVE READY ?
      MOVB      [PP].CI, [GB]           ;MSG SEL TO CI BYTE
                                        ;IN PARAMETER BLOCK
      JMCNE     [PP].CI,RXRDY2          ;0 THRU 7
      MOVI      MC, SIX_SEV_COMPARE     ;MASK COMP   6 OR 7
      JMCE      [PP].CI, MSGSEL         ;6 OR 7 ?
      MOVI      MC, ZERO_COMPARE        ;MASK COMPARE FOR 0
      JMCE      [PP].CI, MSGSEL         ;0 ?
      MOVB      [GB], [PP].CI           ;ECHO
INTR86:         SINTR                   ;INTERRUPT 8086/N
      HLT                               ;WAIT FOR CA
;
DEMO   ENDS
END
```

VITA

Donald E. Palmer was born on Feburary 7, 1950 in Fountain Head, Tennessee. He graduated from White House High School, White House, Tennessee in 1968. Afterward, he attended United Electronics Institute in Louisville, Kentucky and graduated from there in 1970. In June 1970 he entered the U.S. Marine Corps, where he served four years as an Electronic Test Equipment Calibrator and Repairman. In 1977, he graduated from Volunteer State Community College in Gallatin Tennessee, and then attended Tennessee Technological University in Cookeville, Tennessee. He received his Bachelor of Science in Electrical Engineering from Tennessee Technological University in 1979. He was commissioned into the Air Force through Officer Training School in September 1979 and was assigned as a Systems Design Engineer at the Air Force Avionics Laboratory at Wright-Patterson AFB, Ohio. He entered the Air Force Institute of Technology in June 1981.

Permanent Address: Route #1
Cottontown, Tennessee 37048

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: AFIT/GE/EE/82D-52	2. GOVT ACCESSION NO.: AD-A124834
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Design of a Prototype Universal Network Interface Device Using Intel 8086 and 8089 16-Bit Microprocessors	5. TYPE OF REPORT & PERIOD COVERED: Master of Science Thesis
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Donald E. Palmer, 1Lt, USAF	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE: December 1982
	13. NUMBER OF PAGES: 192
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release: IAW AFR 190-17.

LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

4 JAN 1983

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Intel 8086 Microprocessor
Intel 8089 I/O Processor
Local Area Networks
Local Computer Networks
Network Interface
Protocols

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

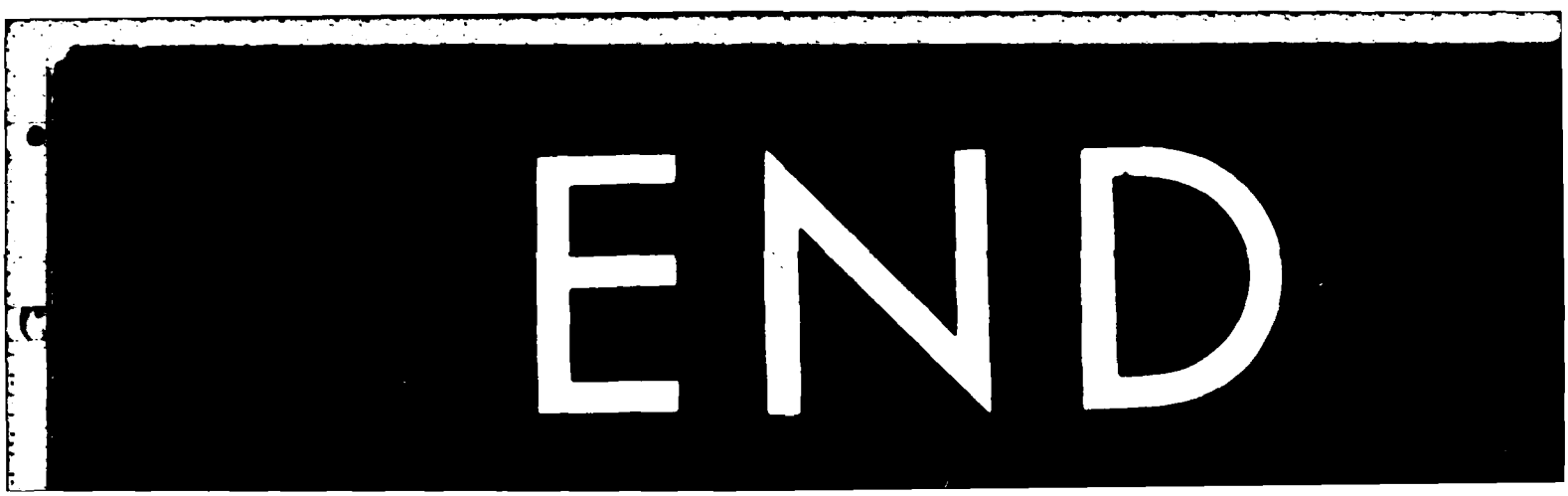
This research describes the development of a Universal Network Interface Device (UNID II) which is intended to function as a network node in a computer communications network. The UNID II is a 16-bit, 8086 microprocessor based version of the present 8-bit 280A UNID being developed at the Air Force Institute of Technology (AFIT). The UNID II's architecture was based on a conceptual block diagram design presented in a previous AFIT thesis. It is

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

comprised of two modules: a local module, which interfaces the UNID II to a host computer and peripheral devices; and a network module, which interfaces the UNID II to a computer communications network. In this report, the detailed design, construction, and testing of the network module is documented. The network module was designed using the Intel 8086 microprocessor family of components, including an 8089 Input/Output processor. An Intel SBC 86/12A single board computer was used as the local module and its testing is also documented. The tests were conducted with the aid of an Intel ICE-86A/88A In-Circuit Emulator. The tests conducted verified the proper operation of the network module's bus interface circuitry, control circuitry, and DMA transfer capabilities. The shared memory, which is used for intermodule communication was also successfully tested. The UNID II was not tested in a computer communications network environment.


END